Estudo de otimização de célula robótica para processo de injeção de volantes

Manuel Baptista Pimenta

Dissertação de Mestrado

Orientador: Paulo Abreu

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Mecânica

Ramo de Automação

Junho de 2017

Resumo

Esta dissertação foca-se na análise e otimização de uma solução robótica existente destinada à automatização de uma célula de injeção de volantes de automóvel. Os desenvolvimentos efetuados ao nível de definição, programação e simulação da célula utilizaram o software RobotStudio, disponibilizado pelo fabricante de robôs industriais ABB.

Definiram-se três áreas de intervenção: otimização do posicionamento dos componentes na célula, a programação da mesma, e o desenvolvimento de uma interface Homem-Máquina para comando e operação da célula.

Foram desenvolvidas três propostas de *layout* da célula, com montagem do robô no solo e em pórtico. Foi também ponderada a utilização de um modelo de robô diferente do inicial.

Na programação da célula foi implementada a produção por lotes, a possibilidade de escolher os moldes a usar e de definir diferentes modelos de volante a produzir.

Foi desenvolvida uma interface para operar a célula, configurar e monitorizar o seu funcionamento.

Por fim foi feita uma análise comparativa entre todas as soluções.

Com as soluções desenvolvidas neste trabalho foi possível obter, relativamente à solução existente, uma redução do tempo de produção de dois volantes de 7%, uma melhor acessibilidade para operações de manutenção e ainda uma interface para a consola do robô que permite selecionar de entre os distintos modelos de volantes e moldes a utilizar.

Optimization analysis of robotic cell for injection of automobile steering wheels

Abstract

This dissertation focus on the analysis and optimization of an existing robotic cell used for the injection of automobile steering wheels. The developments made at defining, programming and simulating the cell used the software RobotStudio, distributed by the manufacturer of industrial robots ABB.

Three main areas of improvements where defined: the layout of the cell, the programming and the development of a Human-Machine interface to control and operate the cell.

Regarding the location of the components within the cell, three different layouts were defined. The use of the robot mounted on a gantry and a new robot model were considered.

In the programming of the robot it was implemented the possibility to choose the molds to operate, to produce in batches and to define different models of steering wheels to inject.

A Human-Machine interface was created to operate the cell, configure and monitor its operation.

At last, a comparative analysis was made between all the developed solutions, as well as with the starting solution.

Using the solutions detailed in this dissertation it was achieved, relative to the initial solution, a gain of 7% in the production time of two steering wheels, and also an improvement in the accessibility of the robotic cell for maintenance operations. Through the human-machine interface, the cell can be easily configured to operate with different molds and models of steering wheels.

Agradecimentos

Agradeço ao orientador desta dissteração, o Professor Paulo Abreu, pela disponibilidade e apoio que demonstrou durante a relaização deste trabalho.

Agradeço à minha família, que sempre me apoiou durante todo o período académico.

Agradeço aos meus amigos, que se demonstraram disponíveis para trocar ideias sobre o tema desta dissertação.

Índice de Conteúdos

Índice de Figuras

Índice de Tabelas

1 Introdução

A automatização dos processos produtivos tem por objetivo a obtenção de um produto final de maior qualidade, repetível, produzido em grandes quantidades, e com um custo associado inferior ao dos produtos fabricados por via manual. No entanto, para tornar uma solução de automação de um processo viável, o seu projeto e conceção devem ser devidamente ponderados e otimizados, sob pena de não se atingir nenhum dos fins à partida definidos, seja por problemas na produção que geram produtos defeituosos, por constantes paragens devidas a falhas no processo ou por custo de operação elevado.

A utilização de robôs industriais na automação de processos produtivos iniciou-se na década de 1960. A primeira grande produtora a incorporar robôs nas suas linhas de produção foi a General Motors, em 1962, com robôs UNIMATE, desenvolvidos pela empresa Unimation. Estes primeiros robôs eram bastante limitados em termos de autonomia e liberdade de movimentos, e eram pensados para executar um pequeno conjunto de tarefas repetitivas, em ambientes fortemente estruturados [1].

O contínuo desenvolvimento da tecnologia associada aos robôs permitiu que a sua gama de aplicações crescesse nos anos seguintes, e a partir de 1980 a produção de robôs começou a ser feita em grande escala. A proliferação dos robôs industriais deveu-se sobretudo à utilização de computadores com poder de cálculo cada vez maior, otimização das estruturas mecânicas dos robôs, incorporação de novos sensores que permitem aumentar a "inteligência" dos robôs, desenvolvimento de aplicações gráficas para a programação, integração em sistemas automáticos de fabrico e interligação com sistemas automáticos de visão.

Atualmente os robôs industriais são aplicados na automação das mais diversas operações, como por exemplo:

- Operações de transporte de materiais, como paletização e alimentação de máquinas;
- Operações de processo, como soldadura por pontos, soldadura por arco elétrico ou pintura;
- Operações de montagem de componentes;

As motivações para a utilização de robôs na automação de processos industriais podem ser várias, como por exemplo condições de trabalho adversas para operadores humanos, tais como temperaturas elevadas, ambientes tóxicos, trabalhos repetitivos, ambientes radioativos. O fator económico também pesa significativamente na decisão de implementação de uma solução robótica.

Nos últimos anos surgiu uma nova categoria de robôs industriais, denominados robôs colaborativos. Esta categoria de robôs distingue-se das restantes por ser especialmente desenvolvida para trabalhar junto de humanos, sem necessidade de barreiras físicas entre o robô e o operador. Para garantir que os operadores não sofrem lesões por acidentes com os robôs, estes estão equipados com múltiplos sensores, tais como sensores de força e sistemas de visão, que lhes permitem parar o movimento antes de causarem danos aos operadores.

Os robôs colaborativos são geralmente compactos, leves e com uma relação capacidade de carga / peso elevada, quando comparados com os robôs industrias comuns. A sua programação é fácil e intuitiva, já que como são dotados de sensores de força, podem ser programados por demonstração, movendo o robô manualmente [2].

Depois desta pequena introdução ao tópico dos robôs industriais, é introduzido o tema desta dissertação de mestrado.

O presente trabalho consiste na análise de uma solução de robotização para uma estação de injeção de volantes de automóvel, e subsequente desenvolvimento de soluções alternativas, com vista a uma otimização do seu funcionamento.

Ao longo desta dissertação são referidas com detalhe as várias etapas pelas quais passou este trabalho, nomeadamente a análise pormenorizada da solução inicial, o desenvolvimento dos novos *layouts* para a célula robótica, as alterações feitas na programação da mesma, a criação de uma interface que permite configurar e monitorizar o funcionamento da célula, e ainda a análise comparativa entre as várias soluções propostas, e entre essas e a solução de partida.

1.1 Enquadramento do projeto e motivação

O ponto de partida deste trabalho foi o projeto de dissertação [3] no qual foi concebida uma célula robótica para automatização de uma estação de injeção de volantes de automóvel. Esta célula robótica devia usar os equipamentos existentes na linha de injeção manual da empresa TRW, na fábrica de Vila Nova da Cerveira. O resultado dessa dissertação foi uma

proposta de *layout* para a célula robótica e desenvolvimento de uma simulação de operação do robô que cumpria com os requisitos inicialmente impostos, nomeadamente tempo de ciclo e funcionamento semelhantes ao processo existente, em que as operações de fabrico eram realizadas por um operador humano.

A motivação desta tese de dissertação foi então dar continuidade ao trabalho efetuado, estudando a possibilidade de melhorar a solução desenvolvida anteriormente, de tal forma que a sua implementação permita ser mais vantajosa para a referida empresa, quer em termos de tempo de ciclo, quer em termos de desenvolvimento de novas funcionalidades de operação da célula, bem como explorar a utilização de outros modelos de robôs industriais disponibilizados pelo mesmo fabricante. Para esse fim recorreu-se à simulação em ambiente virtual da célula robótica, usando o software RobotStudio.

1.2 Objetivos do projeto

Os objetivos desta dissertação são os seguintes:

- Análise da solução existente de *layout* e programação da célula robótica;
- Desenvolvimento de várias propostas alternativas de *layout* da célula robótica, e respetiva otimização das trajetórias;
- Otimização da programação da célula robótica, por forma a permitir a operação com moldes diferentes, e ainda prever a possibilidade de operar apenas com um dos moldes, quando o outro se encontrar em manutenção;
- Desenvolvimento de uma Interface Homem-Máquina que permita configurar e monitorizar o funcionamento da célula robótica;
- Análise comparativa entre as várias soluções propostas e entre estas e a solução de partida, em termos de tempo de funcionamento, atravancamento, funcionalidades de operação.

1.3 Estrutura da dissertação

O presente capítulo introduz o tema da dissertação e apresenta os objetivos da mesma.

No capítulo dois é feita uma análise à solução de partida, proposta pelo colega Rafael Martins na tese de dissertação “Conceção e simulação de um sistema robótico para operações de injeção de volantes” [3].

No terceiro capítulo são desenvolvidos três novos *layouts* de célula robótica alternativos, sendo descrito o trabalho realizado na área da programação da célula robótica e o desenvolvimento da interface Homem-Máquina, para configuração e monitorização da célula robótica.

O capítulo quatro apresenta uma análise comparativa das tês soluções propostas.

Por fim, o capítulo cinco apresenta as conclusões e as perspetivas de trabalhos futuros.

2 Estudo e apresentação do problema

Neste capítulo é analisada a solução existente com o fim de identificar as áreas a serem alvo de otimização. É estudado o posicionamento dos componentes na célula, bem como a programação do robô e as funcionalidades de operação implementadas na sua programação.

2.1 Análise da solução proposta no trabalho anterior

A célula robótica é constituída por um robô ABB IRB 4600-40kg/2.55m, posicionado no solo, ao centro da célula robótica. Relativamente à estrutura mecânica deste robô, é do tipo articulado, com seis juntas de rotação, que lhe conferem seis graus de liberdade. A capacidade de carga do robô são 40 kg, e o seu alcance máximo são 2.55 m.

Montado no órgão terminal do robô encontra-se uma garra desenvolvida para a tarefa em questão. Esta garra possui seis ferramentas:

- garra de três dentes, para manipulação dos volantes e dos aros;
- pistola de pulverização para a laca;
- pistola de pulverização para o desmoldante;
- ponteira de sopro, para limpeza dos canais de injeção;
- agulha de extração de setas;
- ferramenta rotativa de lixagem/polimento, para limpeza dos moldes [3].

A massa total desta garra foi estimada em 6.5 kg, na tese de dissertação que precedeu esta.

Fazem também parte da célula:

- dois moldes semelhantes, M1 e M2;
- um cabeçal de injeção;
- um diapasão;
- dois tapetes de rolos.

Os moldes estão dotados de um sistema de articulações, que permite posicionar o molde em três configurações distintas: posição de fecho, posição de injeção e posição de aberto. O cabeçal também possui articulações de forma a conseguir posicionar o bico de injeção corretamente em relação ao molde. O cabeçal de injeção encontra-se montado numa plataforma, sobre um eixo de translação. Montado na mesma plataforma existe um diapasão, destinado a soltar as setas da ferramenta de remoção das mesmas. os dois tapetes de rolos destinam-se à alimentação dos aros dos volantes, e retirada dos volantes já injetados. A alimentação de aros no tapete é feita por um operário, bem como a retirada dos volantes do fim do tapete de saída.

A Figura 2-1 representa a célula robótica na solução inicial, e permite perceber a disposição dos componentes no espaço. A Figura 2-2 complementa a anterior, com uma vista de cima e algumas cotas relevantes da célula.

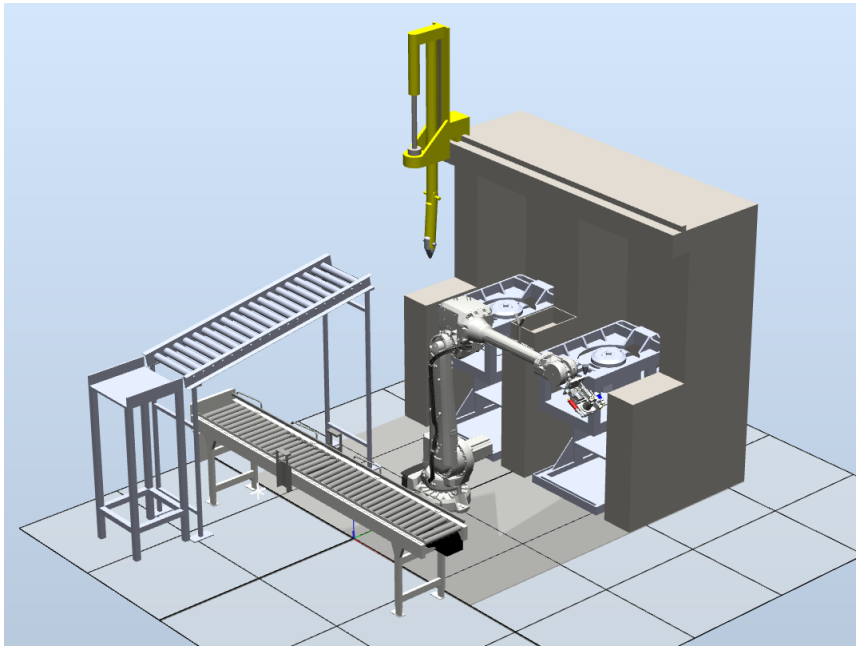


Figura 2-1 - *Layout* da célula robótica inicial

Em termos de funcionamento, a célula foi programada para funcionar sempre com os dois moldes, realizando operações em cada um deles alternadamente. O trabalho inicia-se sempre pelo molde M1.

A injeção dos dois primeiros volantes leva cerca de 378 segundos a ser realizada, e quando em funcionamento cíclico cerca de 306 segundos.

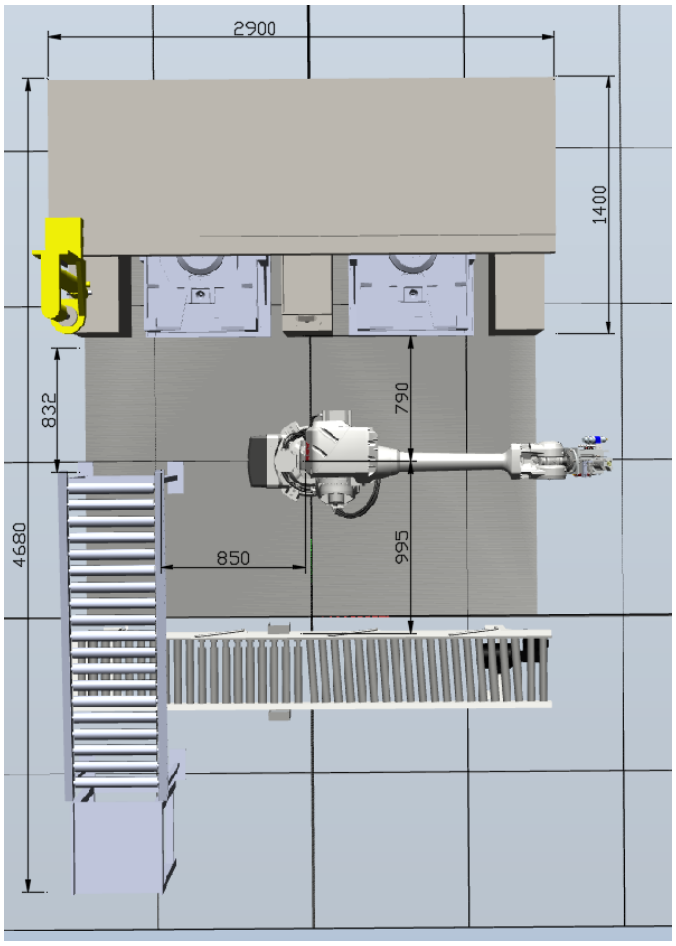


Figura 2-2 - Vista de cima da solução inicial para a célula robótica, com cotas relevantes

Para representar na simulação o comportamento dos componentes auxiliares ao robô, tais como os tapetes transportadores, cabeçal de injeção, moldes e diversas ferramentas da garra, estes foram modelados em sistema CAD e incorporados no ambiente do programa RobotStudio como *Smart Components* (SC). Esta funcionalidade do RobotStudio permite simular movimentos dos dispositivos replicando o seu funcionamento de acordo com procedimentos semelhantes aos que serão utilizados com os dispositivos reais. Para este efeito cada *Smart Component* dispõe de um conjunto de sinais de entrada e de saída que são utilizados quer para alterar o estado desses *Smart Components*, quer para disponibilizar informação acerca do seu funcionamento, sendo usados para a programação da célula robótica

O controlador virtual IRC5 usado para realizar a simulação foi configurado com apenas uma tarefa (*Task*), encarregue de fazer a movimentação do robô e ainda de atuar os vários *Smart Components*.

No que diz respeito à programação, as rotinas criadas tanto para movimentação do robô como para atuação dos SC são executadas sequencialmente, ou seja, uma de cada vez, sem sobreposições. Isto implica que durante certos momentos da produção, o robô esteja parado à

espera que se realize a injeção num dos moldes, por exemplo, quando podia estar a trabalhar no outro molde. Existem vários outros exemplos de situações em que é utilizado este procedimento. A única exceção a este funcionamento sequencial é o SC responsável pela criação e movimentação dos aros de volante no tapete de entrada. Neste caso foi usada a funcionalidade *Interrupts* do controlador IRC5 que permite a implementação da execução de dois processos em simultâneo. Com a utilização desta funcionalidade o aro de volante é criado e movido até ao ponto de *pick-up* sem que o robô tenha de parar o seu movimento.

A sequência de operações realizada pela célula robótica, bem como o tempo de execução de cada operação, é apresentada na Tabela 1.

Tabela 1 – Sequência de operações a executar na célula robótica. Adaptado de [1].

Nº operação	Operação	Tempo de execução	Fase do processo
1	Inicialização	8	Molde 1
2	Aplicação de desmoldante na parte inferior de M1	4	
3	Aplicação de desmoldante na parte superior de M1	8,5	
4	Aplicação de laca na parte inferior de M1	4	
5	Aplicação de laca na parte inferior de M1	8,5	
6	Colocação de aro em M1	12,5	
7	Injeção em M1	25,5	
8	Aplicação de desmoldante na parte inferior de M2	4	Molde 2
9	Aplicação de desmoldante na parte superior de M2	8,5	
10	Aplicação de laca na parte inferior de M2	4	
11	Aplicação de laca na parte inferior de M2	8,5	
12	Colocação de aro em M2	8,5	
13	Injeção em M2	34	
14	Retirar setas em M1	8	Molde 1
15	Retirar volante injetado em M1	15	
16	Limpeza de canais em M1	13	
17	Limpeza da parte inferior do M1	24	
18	Limpeza da parte superior do M1	30	
19	Aplicação de desmoldante na parte inferior de M1	4	
20	Aplicação de desmoldante na parte superior de M1	8,5	
21	Aplicação de laca na parte inferior de M1	4	
22	Aplicação de laca na parte inferior de M1	8,5	
23	Colocação de aro em M1	12,5	
24	Injeção em M1	25,5	
25	Retirar setas em M2	8	Molde 2
26	Retirar volante injetado em M2	11	
27	Limpeza de canais em M2	13	
28	Limpeza da parte inferior do M2	24	
29	Limpeza da parte superior do M2	30	
	Retorno à operação 8		

2.2 Identificação das possibilidades de otimização

A análise à solução focou-se principalmente em três aspetos: disposição dos componentes da célula, programação da mesma, e possibilidade de configuração do seu funcionamento por um operador. Assim, foram identificadas as possibilidades de otimização em cada um desses aspetos.

O robô está posicionado no centro da célula, orientado a 90˚ no sentido horário, se considerarmos a direção dos moldes como a referência, como se pode observar na Figura 2-3. Ora, como a junta 1 do robô tem um intervalo de funcionamento de -180˚ a 180˚, o robô tem de fazer o percurso entre os moldes e o tapete de saída dos volantes movendo-se no sentido horário, o que constitui uma trajetória mais longa do que se movesse no sentido contrário. Por esta razão, um reposicionamento tanto do robô como do tapete de saída de volantes pode conduzir a uma diminuição do tempo de operação.

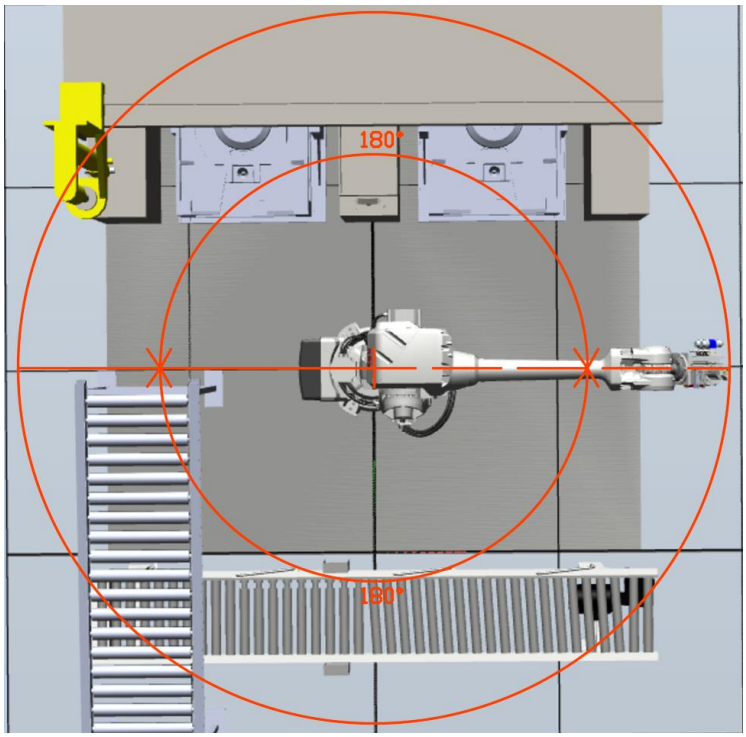


Figura 2-3 - Vista de cima do espaço de trabalho do robô IRB4600 na célula robótica inicial

A revisão do posicionamento do robô poderá conduzir a um ganho em termos de atravancamento da célula, nomeadamente se este for colocado num pórtico. Esta solução permite libertar algum espaço no solo, o que facilitará as trocas de molde e as operações de manutenção. Já quanto ao tapete de entrada, este poderá ser movido para mais perto do posto

de injeção, mantendo a atual orientação, ou então ser desenvolvida uma nova solução de tal forma que este fique paralelo ao tapete de saída.

A posição de descanso do cabeçal de injeção pode ser sujeita a otimização, passando a ser a meio do posto, entre os dois moldes, tendo em vista a redução da distância a percorrer em cada ciclo de injeção, e assim diminuir o tempo global de operação.

Quanto à programação é de notar que as tarefas correm todas em série, ou seja, nunca há mais do que uma tarefa a ser executada ao mesmo tempo. É possível reduzir o tempo de ciclo se algumas das tarefas forem executadas em paralelo. Um exemplo desta situação ocorre quando um volante concluído é colocado no tapete de saída; nessa altura, o robô espera que o volante saía de cena antes de continuar; ou então quando se começa a injeção de um novo volante, o robô está em espera até que seja concluída a injeção, quando podia estar a remover as setas do outro molde.

Em relação à configuração do funcionamento da célula por um operador, a atual solução não contempla esta possibilidade. Foi pensada a criação de uma interface para a consola do robô, que permita a seleção dos moldes em funcionamento, de tal modo que possam funcionar os dois moldes, ou então apenas um dos dois, no caso de estarem a decorrer operações de manutenção. A interface também poderá permitir a escolha do modelo de volante a fabricar, bem como monitorizar o funcionamento da célula.

3 Desenvolvimento e apresentação das soluções

As soluções desenvolvidas procuraram responder às possibilidades de otimização identificadas no capítulo 2, onde se analisou a solução inicial e se discutiram pontos passíveis de melhoramentos.

Foram desenvolvidas três novas soluções para a célula robótica, com diferentes *layouts* dos componentes, por forma a avaliar o impacto do posicionamento destes na performance da célula. Depois de definidos os novos *layouts*, todas as trajetórias que o robô realiza foram ajustadas para acomodar a mudança de localização dos componentes.

O desenvolvimento das soluções foi feito recorrendo ao software de programação e simulação de robôs industriais RobotStudio, distribuído pela ABB.

O RobotStudio disponibiliza bibliotecas de componentes a incorporar nas células, como robôs, mesas posicionadoras, tapetes, ferramentas e eixos lineares [4]. É também possível importar geometrias provindas de programas CAD e incluí-las na simulação.

O software RobotStudio dispõe de modelos virtuais dos controladores dos robôs, pelo que durante a simulação de um qualquer percurso é feita a validação da aplicabilidade do mesmo na célula real [4]. Está também disponível um modelo virtual da consola de programação do robô, e um módulo dedicado ao desenvolvimento de interfaces para esta consola.

Algumas das principais funcionalidades do RobotStudio são:

- Importar geometrias modeladas em programas de CAD, que podem ser mais tarde definidas como peças, ferramentas, ou outros componentes da célula;
- Criação de objetos através das suas funcionalidades de CAD;
- Gerar trajetórias automaticamente a partir de um modelo CAD;
- Detetar colisões e verificar o alcance do robô;
- Utilizar aplicações de software específicas para operações que envolvam maquinagem, soldadura, entre outras;
- Verificar, através de simulação gráfica, as trajetórias programadas;
- Verificar a aplicabilidade do programa na célula real;

- Criação de *Smart Components* [5].

No presente trabalho os *Smart Components* são um componente de grande relevo na criação da simulação, pelo que de seguida se expõe brevemente esta funcionalidade do software RobotStudio.

Um *Smart Component* é um objeto no ambiente RobotStudio, com ou sem representação gráfica, com um comportamento programável. Esta programação pode ser feita em linguagem C#, ou então por agregação de elementos de base disponíveis na biblioteca do RobotStudio [6].

Na criação de um novo *Smart Component*, são normalmente necessárias três etapas:

- Definição da geometria, caso o *Smart Component* tenha representação visual;
- Definição da cinemática;
- Definição do comportamento lógico.

Para proceder à definição da geometria de um *Smart Component* podem seguir-se duas vias: importar o modelo CAD de um ficheiro em formato standard, proveniente de um programa de CAD, ou usar as funcionalidades de CAD disponibilizadas no RobotStudio e criar o modelo pretendido.

De seguida é necessário definir a estrutura cinemática do *Smart Component*, caso se trate de um mecanismo. Para isso, é disponibilizado um menu onde se definem as juntas de translação ou rotação, respetivos limites de funcionamento, e tempos de transição entre as posições extremas dessas juntas.

A última parte da criação de um *Smart Component* é a definição da sua lógica comportamental. Para tal, estão disponíveis um conjunto de sinais de entrada e de saída, que juntamente com uma biblioteca de *Smart Components* de base permitem recriar na simulação o comportamento pretendido. Os sinais de entrada e saída referidos acima são usados na programação da célula robótica, para alterar o estado e para disponibilizar informação acerca do funcionamento dos *Smart Components*.

Alguns exemplos das potencialidades dos *Smart Components* são:

- Movimentação de mecanismos, quer dos disponíveis nas bibliotecas do RobotStudio, quer dos criados pelo utilizador;
- Movimentação de objetos dentro da célula robótica;
- Animação de elementos de segurança das células robóticas, tais como sinalizadores, barreiras de segurança ou semáforos.

- Melhoramento do ambiente gráfico da simulação, sendo alguns exemplos a movimentação programada da câmara ou a criação de partículas de soldadura.

3.1 Solução 1 – IRB4600 no solo

Na solução 1 é utilizado o mesmo robô da solução inicial, ABB IRB 4600-40kg/2.55m, montado no solo. A base do robô foi rodada 90˚no sentido horário, e aproximada do posto de injeção. Alguns dos restantes componentes foram reposicionados, nomeadamente o tapete de alimentação dos aros de volante, que passou a estar no extremo oposto da célula, paralelo ao tapete de saída dos volantes, e o cabeçal de injeção, que viu a sua posição de repouso alterada para o meio do posto de injeção. A Figura 3-1 apresenta uma imagem da célula robótica proposta, enquanto que a Figura 3-2 apresenta a respetiva planta.

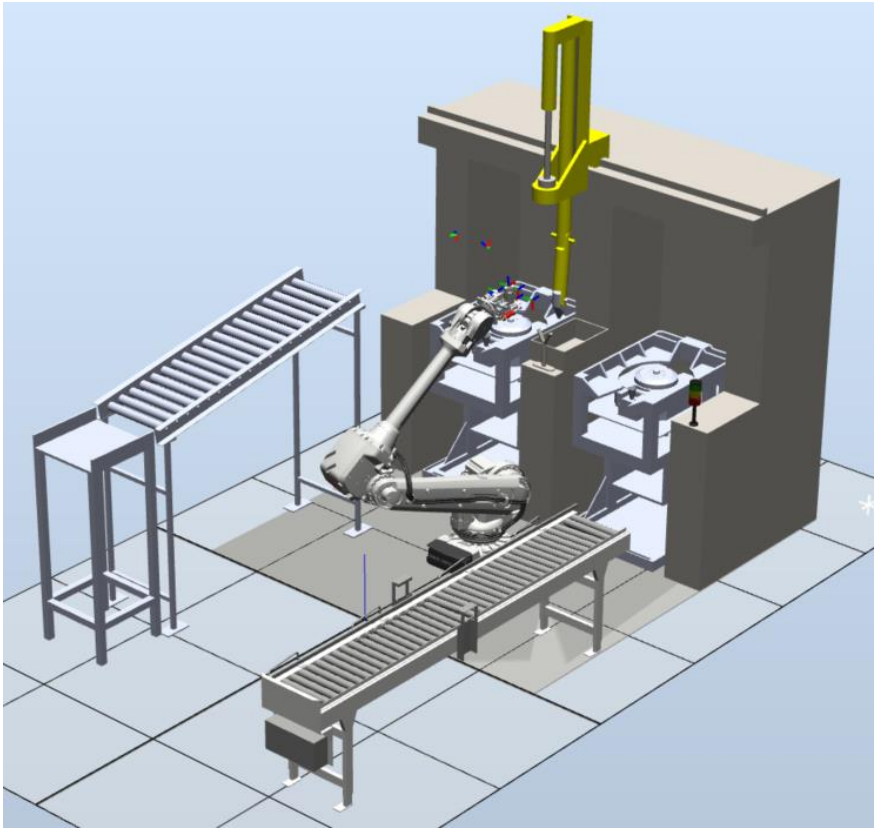


Figura 3-1 - Célula robótica com robô IRB4600 no solo (solução 1)

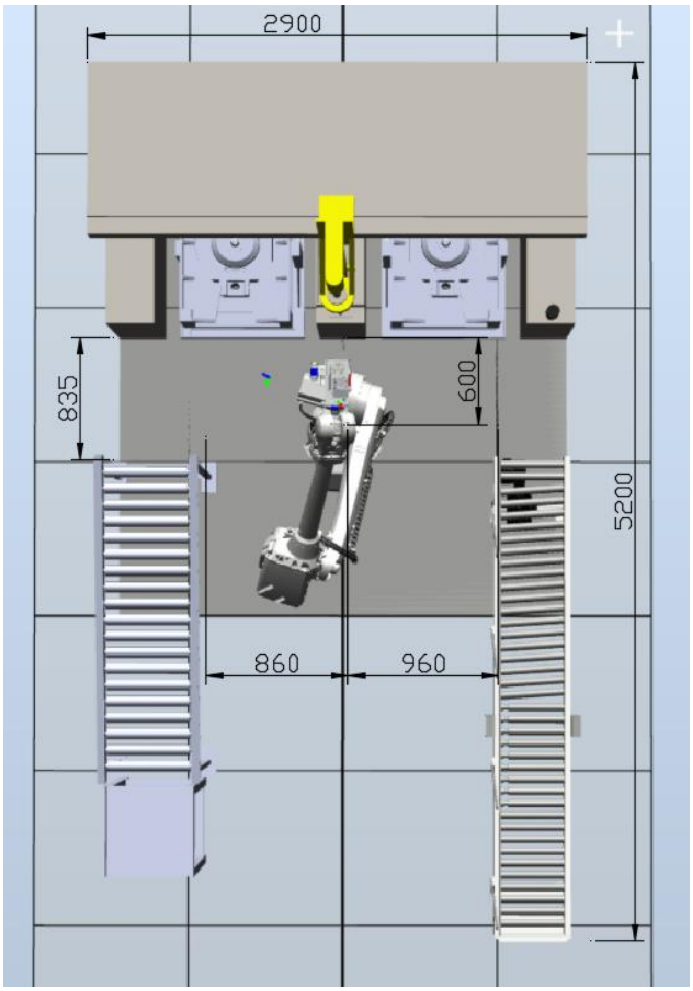


Figura 3-2 - Planta da solução 1, com cotas relevantes

3.2 Solução 2 – IRB4600 em pórtico

As modificações introduzidas na solução 2 partiram da constatação de que na anterior proposta o espaço de trabalho se encontrava excessivamente ocupado pelo robô, o que dificulta operações de manutenção e mudança de moldes. Atendendo a isto, estudou-se a montagem do mesmo robô, ABB IRB 4600-40kg/2.55m, em pórtico e em posição invertida.

Recorrendo às funcionalidades de simulação e validação de alcance disponibilizadas pelo software RobotStudio, foram analisadas distintas opções de posicionamento do robô, tendo-se verificado que a opção de colocar a base do mesmo a 3300 mm do solo seria adequada. Assim, foi pensado um pórtico para permitir fixar o robô a essa cota.

Foi desenvolvido um modelo CAD do pórtico, constituído por dois pilares, sobre os quais é montada a viga que irá suportar o robô. A ligação entre os pilares e a viga é feita por parafusos. Na viga está aparafusada uma placa para a montagem do robô, com três furos alinhados com os furos da base do robô, destinados à fixação do mesmo. O robô é ligado ao pórtico através de

três parafusos. As dimensões e forma deste pórtico são semelhantes às usadas pela empresa Güdel [7] nos pórticos que fabrica e comercializa, especificamente para este modelo de robô.

Os pilares do pórtico ultrapassam os limites da área de implantação das anteriores células robóticas, ou seja, nesta solução o espaço alocado à célula robótica terá de ser superior. Este facto prende-se com a necessidade de colocar os pilares fora do espaço de trabalho do robô, de forma a não interferirem no seu funcionamento. Na Figura 3-3 está representada a solução com o robô montado em pórtico.

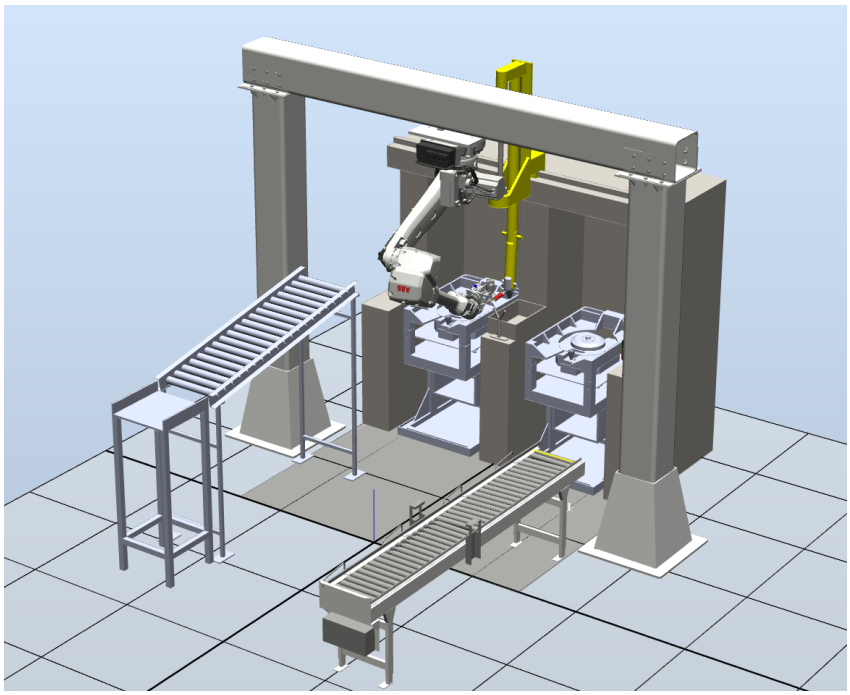


Figura 3-3 - Célula robótica com robô IRB 4600-40kg/2.55m montado em pórtico invertido (solução 2)

A Figura 3-4 representa a célula robótica vista de cima, com algumas cotas relevantes. A área de implantação desta solução tem um atravancamento total de 5,2 m por 4,6 m.

Foi definida uma posição para o robô que minimiza a sua interferência quando for necessário realizar operações de manutenção ou mudança de moldes na célula, e que pode ser vista na Figura 3-5. A distância livre ao solo nesta posição é de 2200 mm.

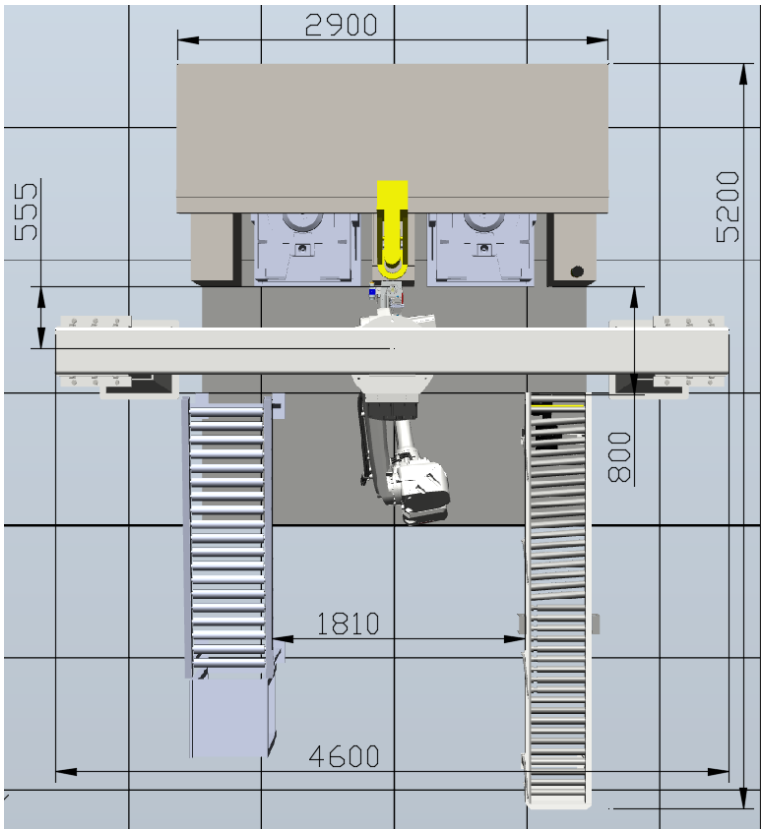


Figura 3-4 – Planta da solução 2, com cotas relevantes

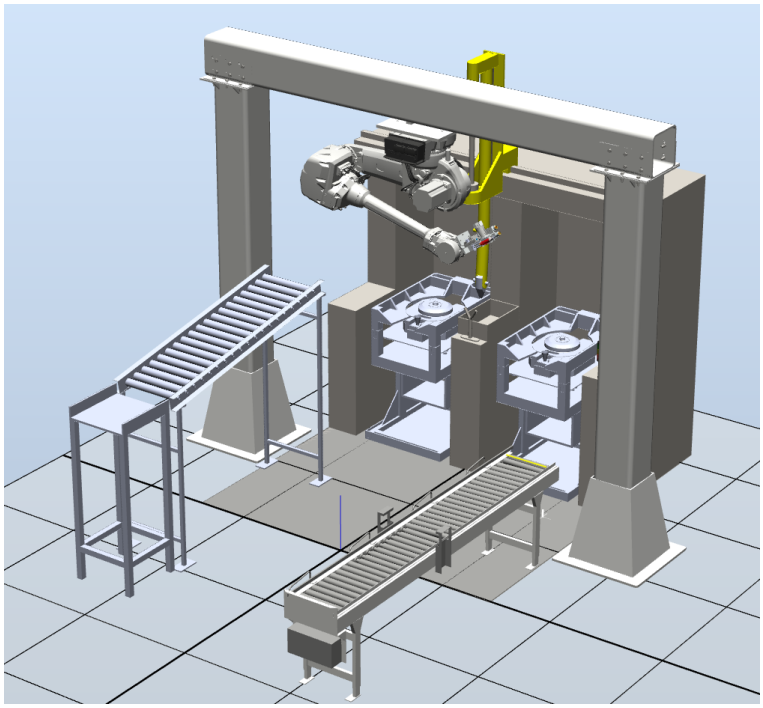


Figura 3-5 – Robô IRB4600 na posição de realização de manutenção ou mudança de molde

3.3 Solução 3 – IRB2600 em pórtico

Com a solução 3 pretendeu-se avaliar a possibilidade de usar um robô de menor capacidade de carga para operar a célula robótica. Tendo em conta que a garra desenvolvida para o robô tem uma massa estimada de 6.5 kg, muito inferior à capacidade de carga do robô IRB 4600-40kg/2.55m, que é 40 kg, procurou-se um robô com capacidade de carga menor.

Foi considerado que em conjunto, a massa da garra e do volante nunca ultrapassaria 11 kg, pelo que se definiu esse valor como o limite inferior para a capacidade de carga do robô a selecionar. Depois de analisar o catálogo de robôs da ABB, foi selecionado o robô IRB 2600ID-15/1.85. Este robô tem uma capacidade de carga de 15 kg e um alcance de 1.85m. De referir ainda que este robô incorpora a tecnologia *Integrated Dressing*, que consiste na passagem de cabos ou tubagens, necessários para o funcionamento das ferramentas, pelo interior do órgão terminal do robô, de forma a que sejam minimizados os esforços nos cabos provenientes da movimentação do robô. Desta forma, o tempo de vida dos cabos e tubagens é aumentado. Além disso, uma vez que os cabos e tubagens se encontram firmemente ligados ao robô, é possível usar as máximas acelerações que o robô permite sem preocupações com cabos pendurados a balançar [8].

Na Figura 3-6 está representado o espaço de trabalho deste robô, em comparação com o robô IRB 4600-40/2.55, utilizado nas anteriores soluções.

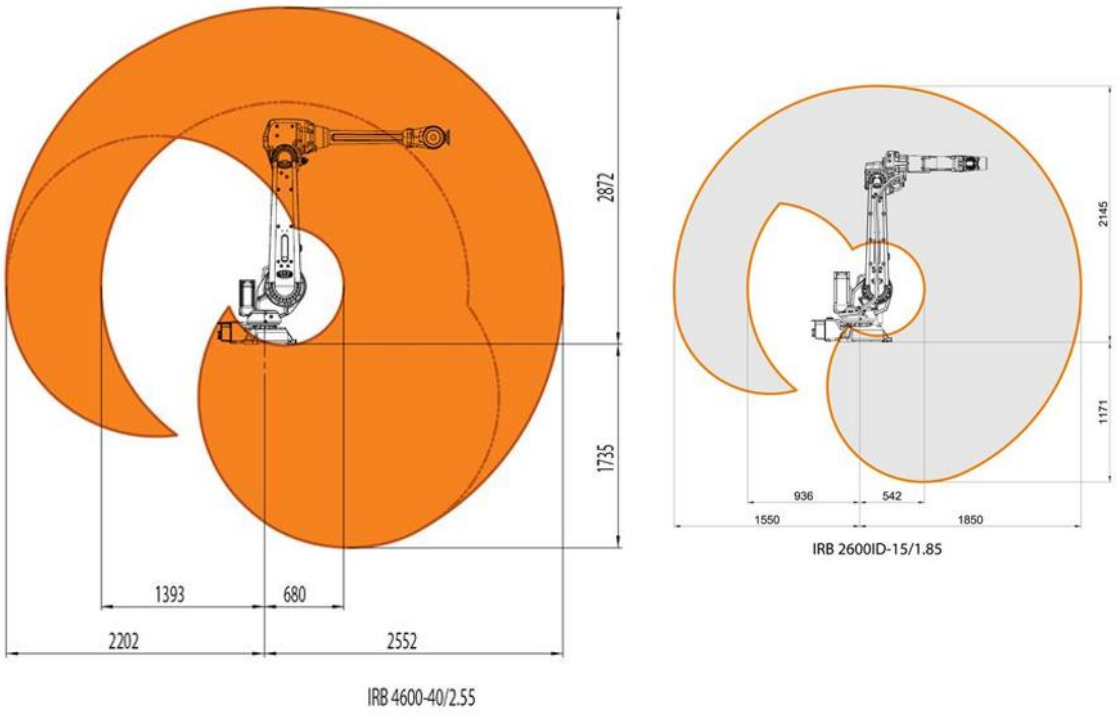


Figura 3-6 - Espaço de trabalho do robô IRB 2600-15/1.85. Adaptado de [9] e [10]

Uma vez que o alcance deste robô é menor que o do usado anteriormente, foi necessário explorar outras alternativas de fixação do robô no pórtico, e de altura do pórtico. O robô foi montado no pórtico com a base ligada à lateral do pórtico, virada para o posto de injeção, ao invés de na parte inferior do mesmo. A altura do pórtico foi diminuída para 2415 mm.

O tapete de entrada de aros viu a sua altura aumentada em 300 mm, para facilitar a chegada ao ponto de *pick-up* dos aros, e o tapete de saída teve a sua altura diminuída em 400 mm, de forma a ser acessível por baixo do pórtico. Por fim, o diapasão, componente utilizado para soltar as setas da agulha de remoção, foi reposicionado e instalado um para cada molde, sendo que um foi posicionado à esquerda do molde M1 e o outro à direita do molde M2. Esta alteração foi necessária já que o robô não conseguia executar as trajetórias necessárias quando existia apenas um diapasão, entre os moldes.

A Figura 3-7 mostra a célula robótica proposta na solução 3, com o robô IRB 2600ID na posição de descanso.

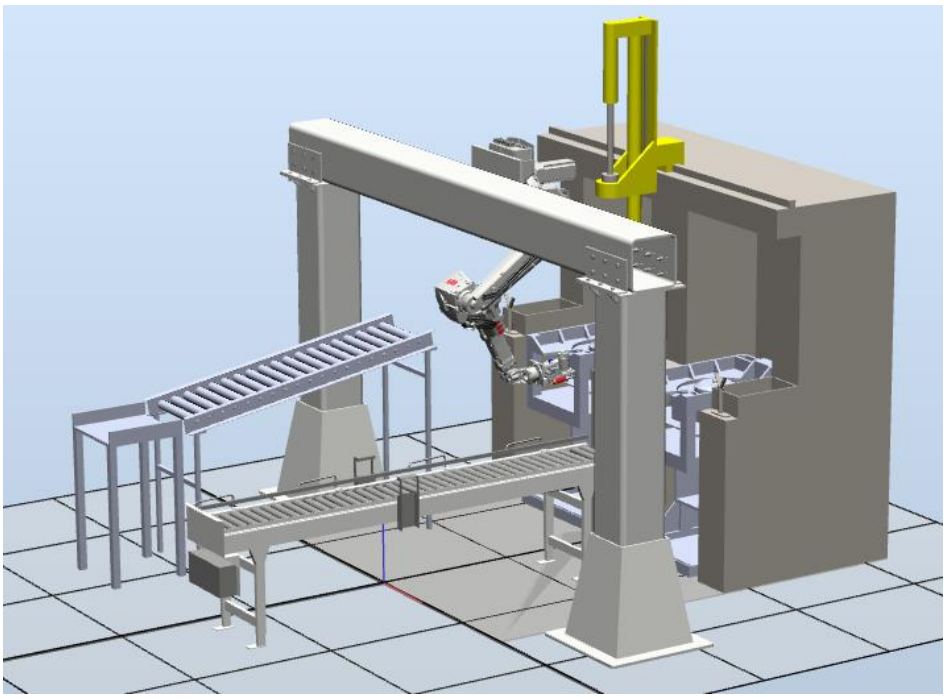


Figura 3-7 – Célula robótica com robô IRB 2600ID-15/1.85 montado em pórtico (solução 3)

Como se pode ver na Figura 3-8, nesta solução o robô ficou mais afastado do posto de injeção do que nas anteriores propostas. Esta posição do robô foi escolhida usando as funcionalidades de simulação e validação de alcance do software RobotStudio, já que uma posição de montagem idêntica à das propostas anteriores não permitia que o robô executasse os percursos necessários.

O atravancamento da área de implantação desta solução mantém-se nos 5,2 m por 4,6m, tal como na solução 2.

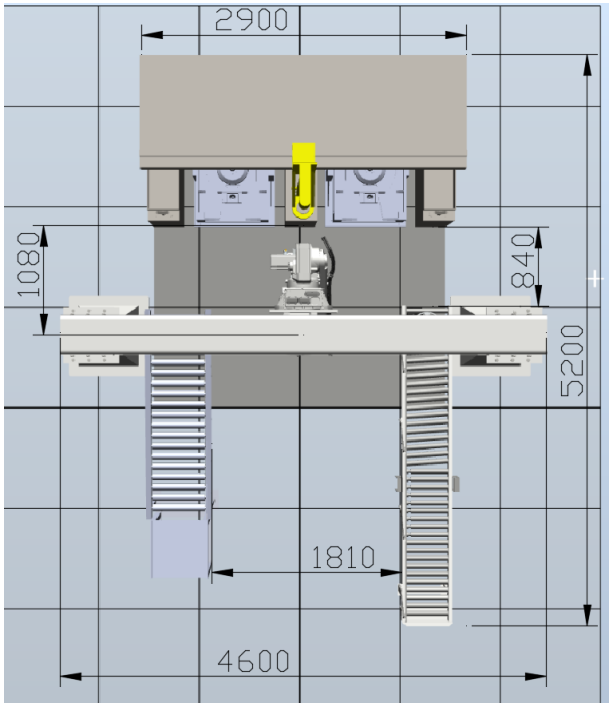


Figura 3-8 – Planta da solução 3, com cotas relevantes

Tal como na solução 2, foi também definida para a presente solução uma posição designada “recolhido”, na qual o robô deve ser colocado quando se pretende realizar operações de manutenção e mudança de molde. Esta posição é apresentada na Figura 3-9. Com o robô nesta posição o espaço de trabalho fica desimpedido, facilitando as referidas operações. A altura livre ao solo na posição de “recolhido” é de 2250 mm.

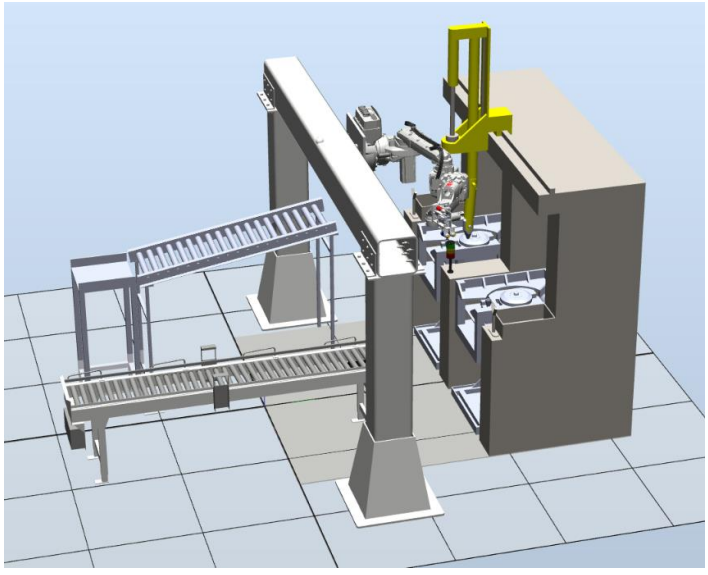


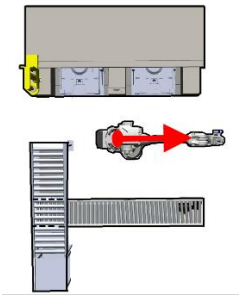
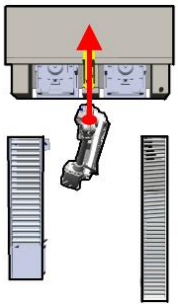
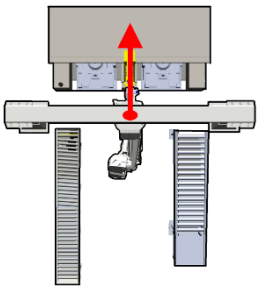
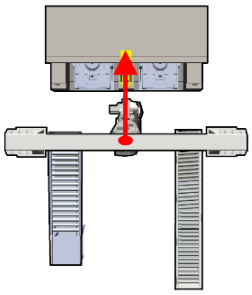
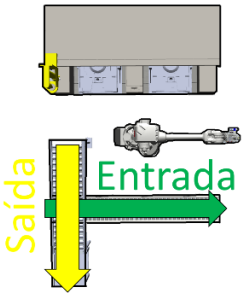
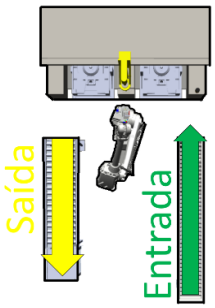
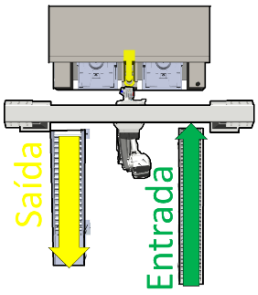
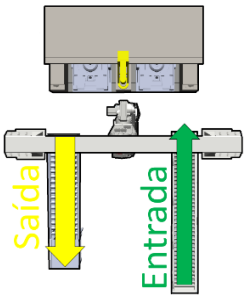
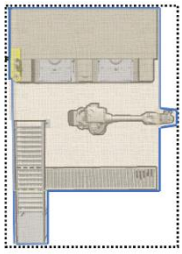
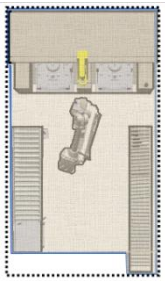
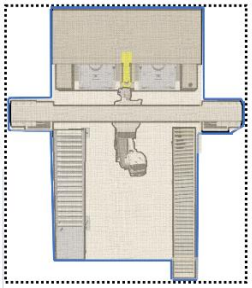
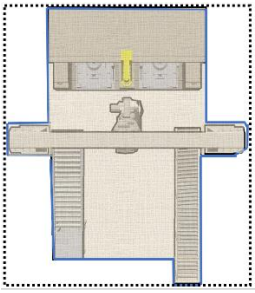
Figura 3-9 - Robô IRB2600ID na posição de realização de manutenção ou mudança de molde

3.4 Resumo dos *layouts* propostos

Esta secção tem como propósito reunir as características mais importantes de todos os *layouts* vistos até aqui, para facilitar a visualização das diferenças existentes e do processo evolutivo que esteve na origem das sucessivas propostas de solução.

Na Tabela 2 foram agrupadas as características mais relevantes de todas as soluções propostas e da solução de partida.

Tabela 2 – Resumo das características das diferentes soluções propostas

	Solução de Partida	Solução 1	Solução 2	Solução 3
Robô	IRB 4600-40/2.55	IRB 4600-40/2.56	IRB 4600-40/2.57	IRB 2600ID-15/1.85
Alcance (m)	2,55	2,55	2,55	1,85
Capacidade de carga (kg)	40	40	40	15
Orientação robô				
Robô em pórtico	Não	Não	Sim	Sim
Orientação entrada/saída produtos				
Posição cabeçal de injeção	Extremo	Central	Central	Central
Área de implantação	 2,900m x 4,680m = 13,6 m^2	 2,900m x 5,200m = 15,0m^2	 4,600m x 5,200m = 23,9 m^2	 4,600m x 5,200m = 23,9 m^2

3.5 Otimização da programação da célula robótica

Ao analisar a solução inicial ficou claro que grande parte da otimização passível de ser realizada passava por uma programação do robô mais eficiente. Nessa solução, as rotinas associadas à movimentação do robô e à manipulação dos *Smart Components* eram asseguradas por apenas uma tarefa, denominada T_ROB1. Ora, como a execução do programa pelo controlador IRC5 é feita linha a linha, e devido à programação adotada, não existiam tarefas em execução em paralelo, o que comprometia o tempo de execução da produção dos volantes. Por exemplo, quando o robô retirava um volante já concluído do molde, e o colocava no tapete de saída, aguardava na mesma posição que o volante percorresse o tapete e saísse da simulação, para só depois continuar o seu funcionamento. Outro exemplo: durante a injeção de um volante o robô esperava na posição de descanso que se concluísse a injeção, que o cabeçal regressasse à sua posição inicial e que o molde voltasse à pose original, quando podia começar a remover as setas do outro molde logo que se iniciasse a injeção.

De forma a estruturar convenientemente a programação da célula robótica começou-se por idealizar a arquitetura de controlo que lhe estaria associada na implementação real. Na Figura 3-10 está representada esquematicamente a estrutura de controlo da célula robótica.

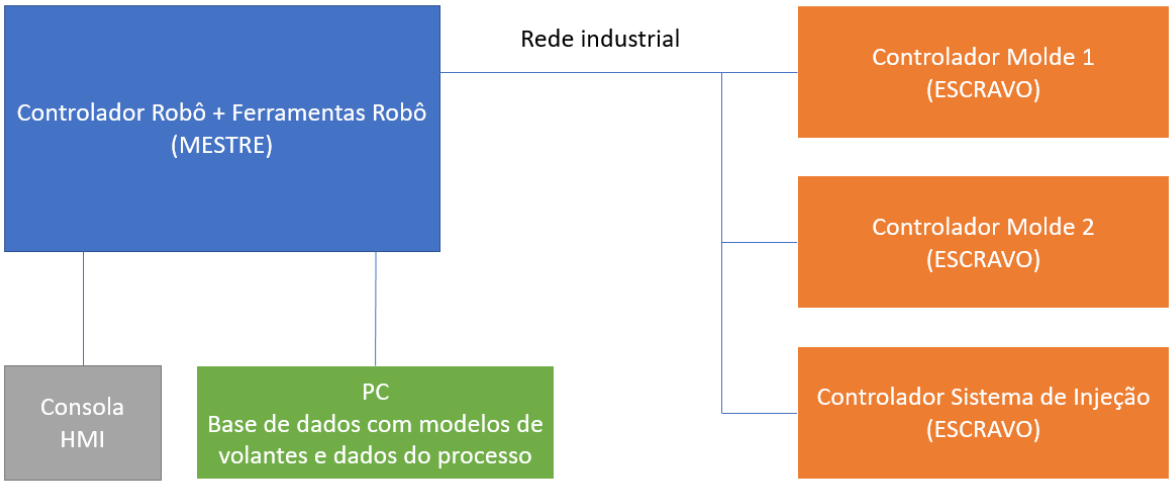


Figura 3-10 - Arquitetura de controlo da célula robótica para implementação real

O controlador do robô, responsável pela movimentação do mesmo e pelo acionamento das ferramentas da garra, funciona como mestre do sistema, enquanto que os restantes componentes são escravos, controlados cada um pelo seu autómato. Existe ainda uma interface Homem-Máquina, materializada pela consola do robô, onde se pode configurar o funcionamento do robô e operar manualmente tanto o robô, como os moldes e o sistema de

injeção. O tapete de saída dos volantes não é referido na Figura 3-10 porque não tem nenhum elemento controlável, já que funciona por gravidade.

Uma vez definida a estrutura básica de controlo da célula real, foi desenvolvida a programação no software de simulação RobotStudio de forma a replicar o funcionamento real pretendido.

Para simular a existência de vários controladores a correr simultaneamente, foram estudadas as funcionalidades do controlador IRC5 *Multitasking e Interrupts.*

A funcionalidade de *Multitasking* é um módulo opcional que se pode adicionar ao controlador, e que permite a criação de várias tarefas, que são executadas simultaneamente. Esta funcionalidade tem grande utilidade por exemplo em células com mais de um robô, sendo que cada robô é controlado por uma tarefa [11].

Um I*nterrupt* é um evento despoletado por uma condição programável, por exemplo, um sinal de I/O. Uma vez detetada a *Interrupt*, a execução da *Trap Routine* a ela associada é iniciada, permitindo ter dois processos a correr em paralelo. As *Trap Routines* são rotinas destinadas a lidar com as interrupções [11].

Depois de analisado o funcionamento de cada uma das funcionalidades referidas, optou-se por usar ambas para a programação da célula. Como se pode ver na Figura 3-11, foi atribuída uma tarefa da simulação em RobotStudio ao robô, uma a cada molde e uma ao sistema de injeção. Desta forma, é obtida uma organização semelhante à que existirá no sistema real.

De notar que neste esquema são contemplados os tapetes de entrada de aros e saída de volantes, uma vez que na simulação estes dois componentes são representados por *Smart Components*, que têm de ser controlados para o correto funcionamento da simulação. O acionamento do tapete de entrada de aros foi incluído na tarefa de controlo do robô, através de uma interrupção, para que o seu acionamento não cause a interrupção da movimentação do robô. O *Smart Component* do tapete de saída de volantes é controlado pela tarefa T_SC, que também controla o sistema de injeção.

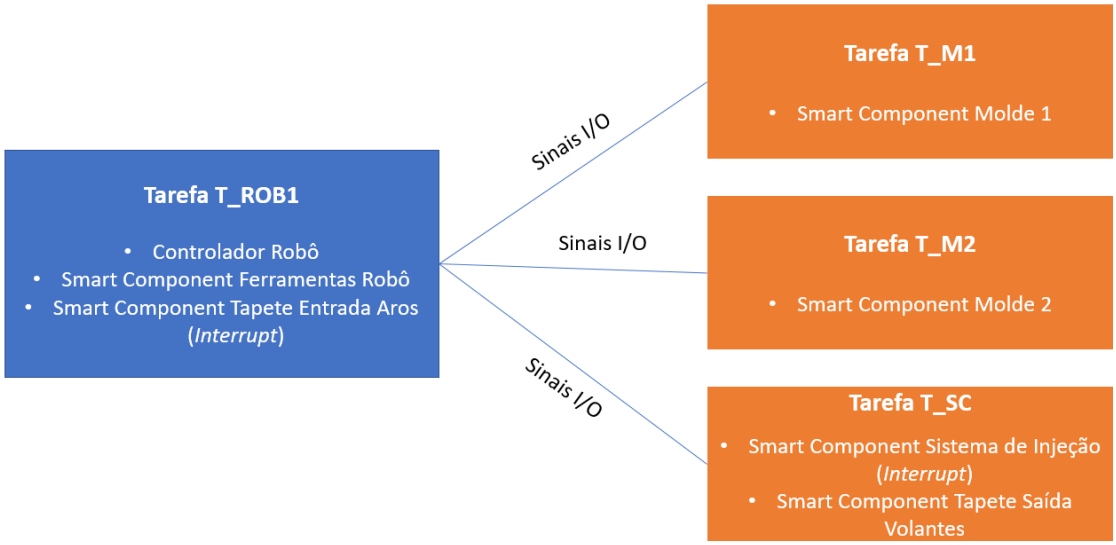


Figura 3-11 - Esquema da estrutura usada na programação da célula robótica

Para a implementação da simulação em ambiente do RobotStudio foi necessário configurar e interligar os distintos S*mart Components* associados a cada um dos equipamentos da célula robótica. Na Figura 3-12 é apresentada a arquitetura global da interação definida para os distintos S*mart Components* e a sua ligação com o controlador do robô.

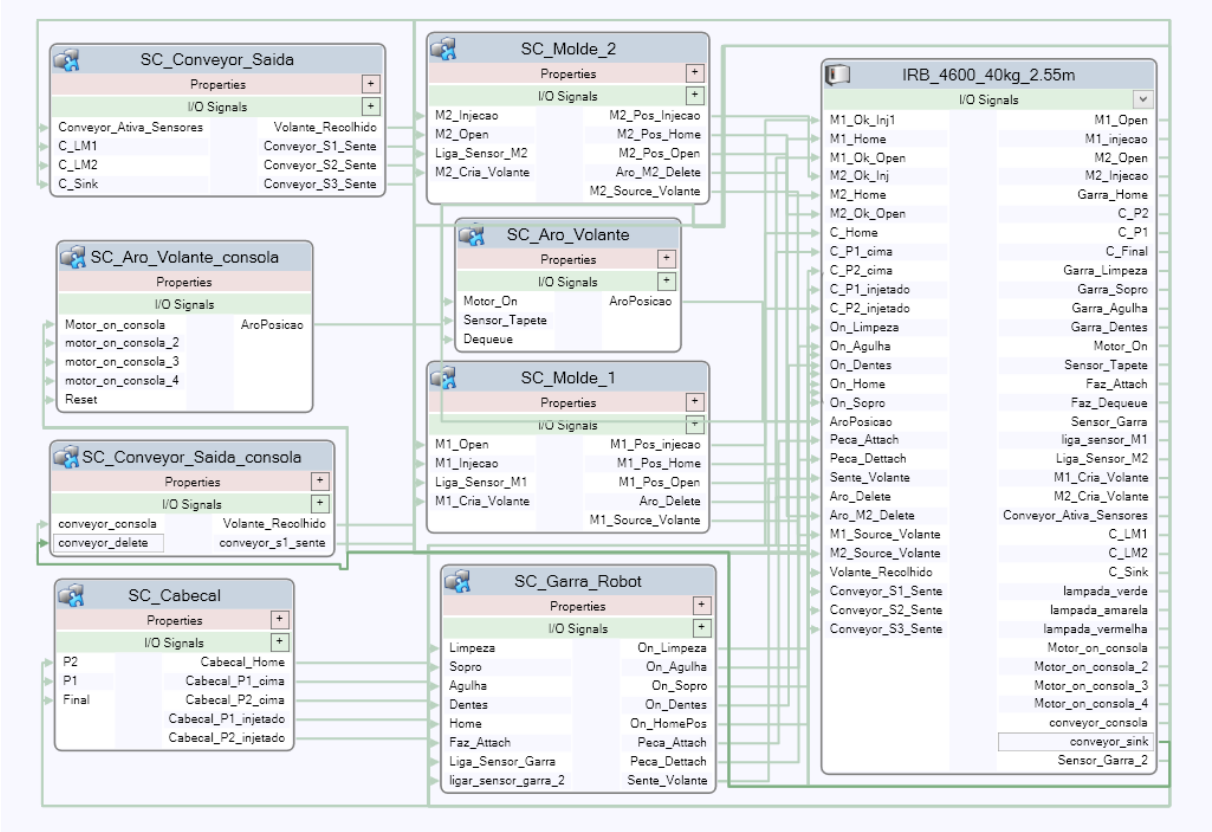


Figura 3-12 - Interligação entre S*mart Components* e o controlador do robô

O controlador foi configurado para dispor da funcionalidade *Multitasking*, sendo necessário ativar essa opção nas configurações do controlador, conforme ilustrado na Figura 3-13.

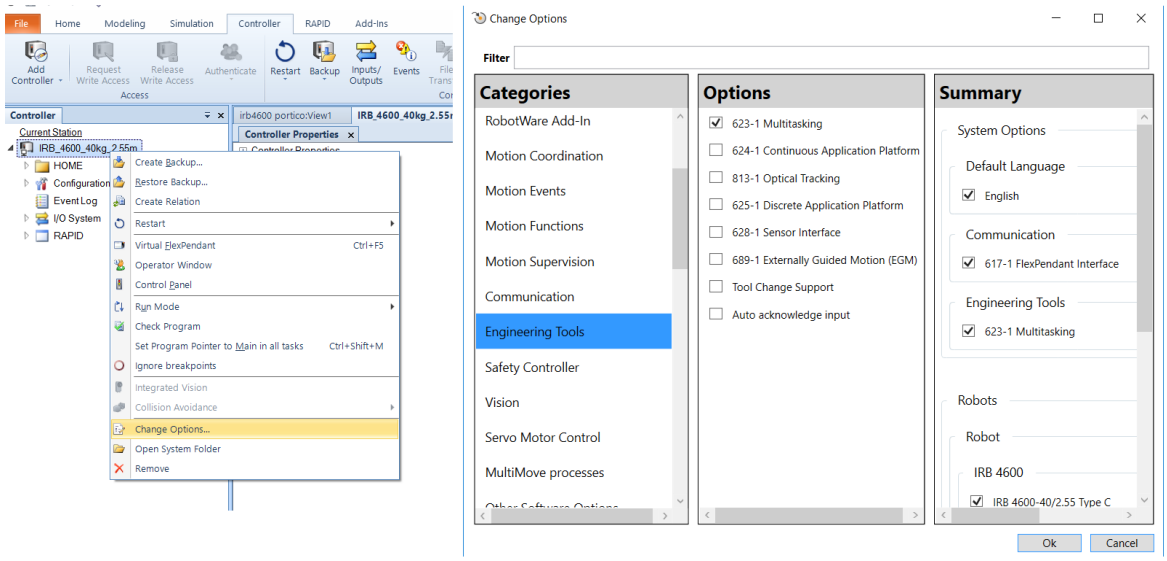


Figura 3-13 - Configuração do controlador IRC5 com o módulo Multitasking

De seguida foram configuradas três novas tarefas para além da tarefa pré-definida T_ROB1, criada aquando da inicialização da estação. As tarefas criadas, assim como as suas finalidades são:

- T_M1, destinada a operar o molde M1;
- T_M2, destinada a operar o molde M2;
- T_SC, destinada a operar o cabeçal de injeção e o tapete de saída dos volantes.

O menu de configuração das tarefas está representado na Figura 3-14.

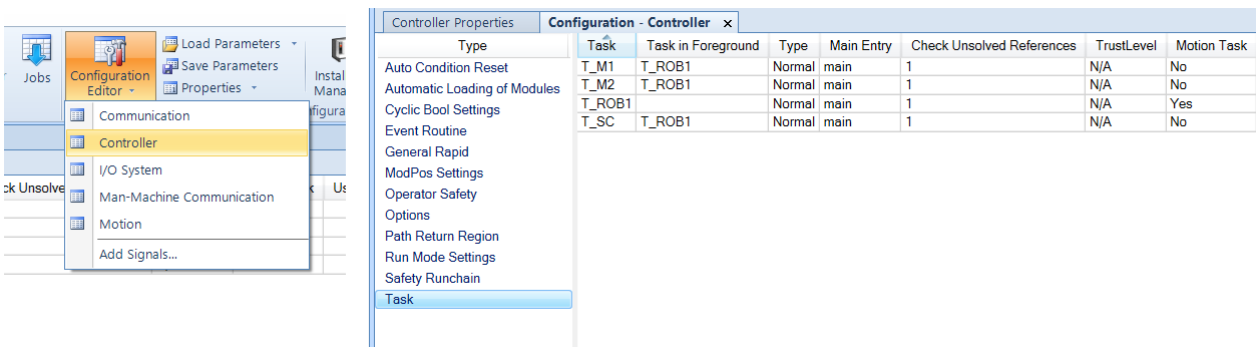


Figura 3-14 – Configuração das tarefas no controlador IRC5

A sincronização entre as várias tarefas é feita através de sinais de I/O, ativados pela tarefa do robô quando é necessária a execução de alguma rotina a cargo das outras tarefas. A ativação desses sinais é detetada pelas restantes tarefas, o que despoleta a execução das rotinas de que estão encarregues.

A funcionalidade *Interrupts* foi usada na tarefa T_ROB1, para a criação e movimentação dos aros de volantes. Este procedimento já estava implementado na solução inicial. As *Interrupts* foram também usadas na tarefa T_SC, sendo criadas duas rotinas *Trap*, uma para comandar o cabeçal durante a injeção no molde M1 e outra para a injeção no molde M2.

Quanto à possibilidade de configuração do funcionamento da célula, foi implementada uma forma de escolher quais os moldes que vão operar, quer através da consola, quer através da alteração de uma instrução RAPID. Para o efeito foram criadas quatro rotinas na tarefa T_ROB1, nomeadamente:

- Molde1, que faz a célula operar apenas com o molde M1;
- Molde2, que faz a célula operar apenas com o molde M2;
- Molde1_2, que faz a célula operar com ambos os moldes, começando pelo molde M1;
- Molde2_1, que faz a célula operar com ambos os moldes, começando pelo molde M2;

Relativamente à possibilidade de produzir diferentes modelos de volante, foi implementada a possibilidade de configurar vários parâmetros característicos de cada modelo. Como se pode ver na Figura 3-10, na arquitetura de controlo da célula robótica real, é proposto os dados dos modelos de volante serem armazenados numa base de dados da linha de produção. Para replicar essa situação na simulação foram criadas variáveis no código RAPID para armazenar os ditos dados. Para editar as características de cada modelo, e criar novos modelos, é necessário modificar essas variáveis que contêm as informações dos modelos, identificadas na Figura 3-15.

```
!Parâmetros dos modelos de volante
PERS string modelos_nome{6}:=["JoyotaTS765","RiatYU6","HeugeotK9i7","GcLarenHJT54","Não Definido","Não Definido"];
PERS string modelos_path_limp{6}:=["PTH34","PTH34","PTH34","PTH34","Não Definido","Não Definido"];
PERS string modelos_path_laca_desm{6}:=["PTH5","PTH5","PTH5","Não Definido","Não Definido"];
PERS num modelos_tempos_cura{6}:=[50,45,50,55,0,0];
PERS num modelos_tempos_inj{6}:=[5,5,5,5,0,0];
```

Figura 3-15 – Variáveis no código RAPID que contêm os parâmetros dos modelos de volante

Os parâmetros considerados na definição de um modelo de volante foram o percurso de limpeza, de aplicação de desmoldante e de laca, o tempo de cura e o tempo de injeção. A

implementação da funcionalidade de alteração de percursos do robô consoante o modelo de volante a produzir foi pensada de tal forma que facilita a adição de novos modelos. Foram criados *templates* para a criação de percursos para novos modelos de volante, cuja estrutura deve ser seguida para futuras expansões do número de modelos.

Outra melhoria implementada na programação foi a possibilidade de a produção ser feita por lotes. No modo de produção por lotes é definindo um número de volantes a produzir, e quando se atingir o número pretendido a célula parará o seu funcionamento, sem que nenhum volante fique a meio da produção e deixando os moldes limpos, prontos para iniciar o funcionamento quando solicitado. A produção por lotes pode ser ativada na interface da consola.

Foi também desenvolvido o código necessário para executar uma paragem controlada da produção, a pedido do utilizador. À semelhança do que foi explicado no parágrafo anterior, para a paragem depois de concluído um lote, quando é solicitada a paragem controlada os volantes já em produção são terminados, e não são introduzidos mais volantes no circuito de produção. Os moldes são deixados limpos e preparados para uma próxima utilização.

De forma a que os tempos de cura sejam respeitados, especialmente na produção do primeiro volante após inicialização, são utilizados temporizadores, um para cada molde. Os temporizadores são iniciados quando se termina a injeção no molde correspondente, com a instrução *ClkStart*, e quando é chegada a altura de o robô retirar as setas desse molde é tirada uma leitura do temporizador, com a instrução *ClkRead* [12]. Essa leitura é comparada com o tempo de cura, definido previamente, e é calculado o tempo de espera que o robô deve cumprir, sendo depois dada a instrução de aguardar esse tempo de espera. O temporizador é depois parado, e posto a zero, para ser utilizado no ciclo de produção seguinte.

3.6 Desenvolvimento de interface para a consola do robô

Com as modificações introduzidas na programação da célula, a mesma passou a ter alguns parâmetros configuráveis, pelo que se tornou relevante a existência de uma interface Homem-Máquina (HMI) que permita configurar e monitorizar o seu funcionamento.

Os propósitos da interface desenvolvida são principalmente três:

- Operar a célula robótica;
- Configurar o seu funcionamento;

- Monitorizar o seu funcionamento.

A HMI foi desenvolvida para a consola do robô, denominada *FlexPendant*, que dispõe de um ecrã tátil, joystick e botões, e que permite controlar a operação de um sistema robótico. Nomeadamente, permite iniciar a execução de programas, fazer *jogging* do robô, modificar programas, ensinar novos *targets*, ajustar a posição de *targets* já existentes, correr interfaces personalizadas para uma determinada aplicação, entre outros.

Para proceder ao desenvolvimento da HMI foi utilizado a ferramenta do RobotStudio chamada *ScreenMaker*, cujo intuito é precisamente a criação de interfaces para a consola *FlexPendant*. De notar que o *ScreenMaker* só está disponível na versão de 32-bit do RobotStudio [13].

A programação no ambiente de desenvolvimento *ScreenMaker* é muito semelhante à programação em *Visual Basic*. É disponibilizada uma caixa de ferramentas onde estão disponíveis os elementos a integrar nos ecrãs da interface, tal como botões, caixas de verificação, caixas de combinação, sinalizadores, entre outros. As ações associadas a cada um dos elementos usados é configurável, e podem ser:

- Fechar ou abrir ecrãs da interface;
- Ler ou alterar valor de sinais do controlador;
- Ler ou escrever variáveis do código RAPID;
- Ler ou escrever variáveis da interface.

O desenvolvimento da interface foi feito usando estes elementos, e configurando as suas ações de forma a corresponder ao funcionamento pretendido para a interface.

O ecrã inicial da interface dá a opção de escolher entre dois modos: Modo Automático ou Modo Manual (Figura 3-16). O Modo Automático é destinado ao funcionamento normal da célula, sendo executado o código RAPID presente no controlador. O Modo Manual permite posicionar o robô em pontos relevantes para o funcionamento da célula, operar os moldes e sistema de injeção, e comandar a garra do robô.

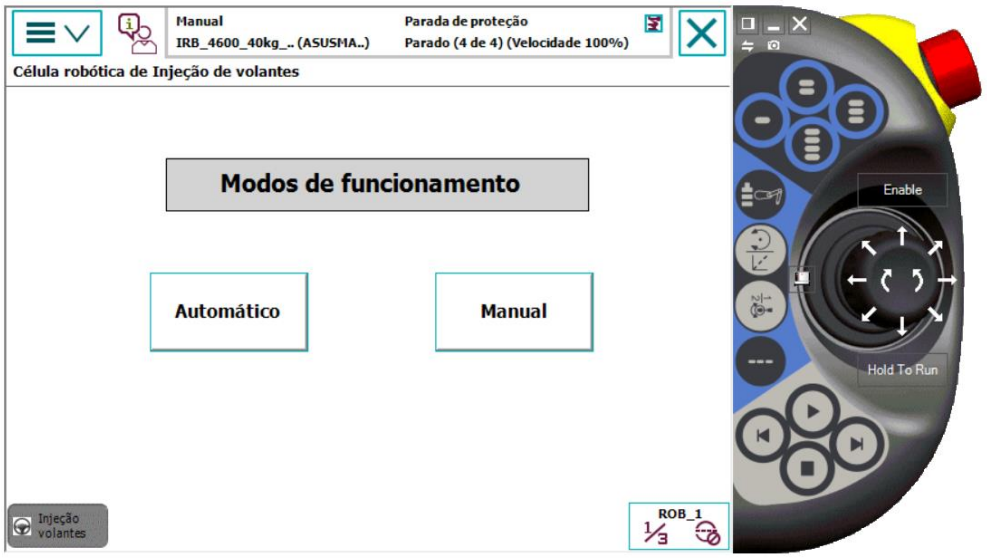


Figura 3-16 – Ecrã inicial da interface desenvolvida para a consola *FlexPendant*

Selecionando o Modo Manual surgem cinco opções, como mostrado na Figura 3-17:

- Comandos Garra;
- Movimentação Robô;
- Comandos Molde 1;
- Comandos Molde 2;
- Comandos Cabeçal Injeção.

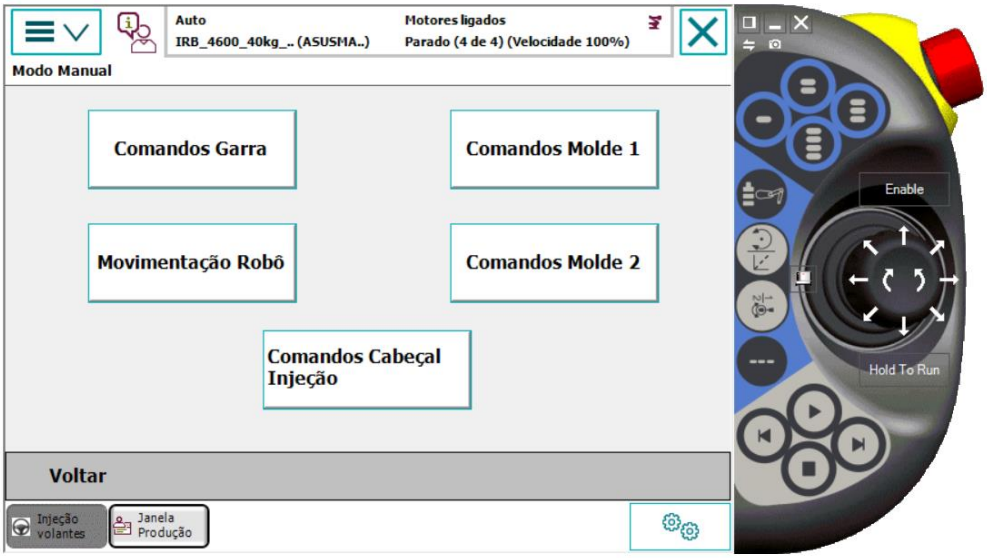


Figura 3-17 – Ecrã do modo manual

No menu Comandos Garra existe um botão para ativar cada ferramenta. A principal utilização destes comandos é a verificação manual do funcionamento das ferramentas, antes de

iniciar o funcionamento da célula. Estão também disponíveis indicadores luminosos que sinalizam o estado de ativação de cada ferramenta (Figura 3-18).

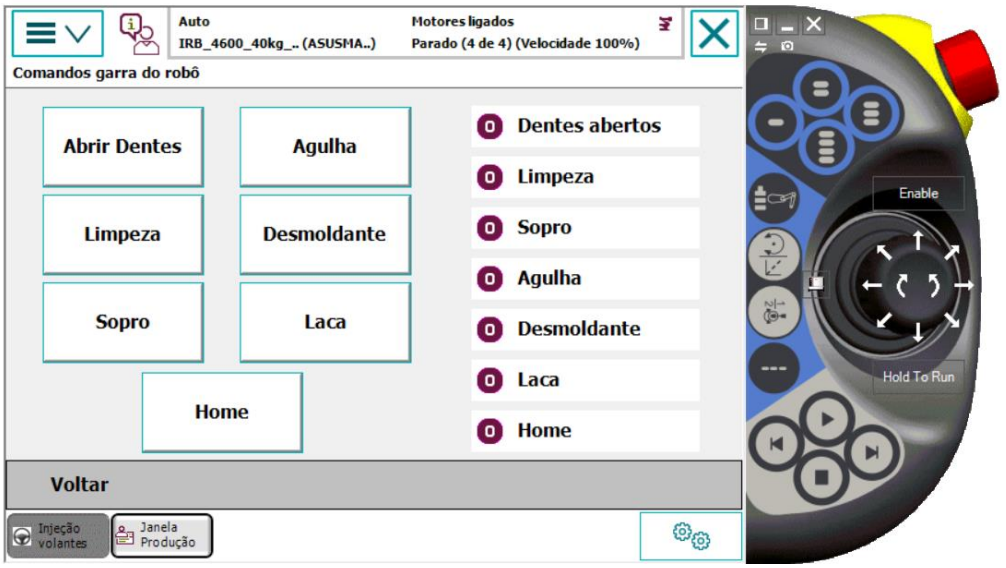


Figura 3-18 – Ecrã com comandos manuais da garra do robô

No menu Movimentação Robô estão disponíveis botões que permitem levar o robô a *targets* relevantes para o funcionamento da célula, como a posição de descanso do robô, a posição "recolhido", que facilita a realização da manutenção dos moldes, entre outros. O objetivo deste menu é permitir reposicionar o robô depois de uma paragem não programada, para continuar os procedimentos interrompidos. Para usar esta funcionalidade é necessário levar o robô para perto do target pretendido, usando o joystick da consola, e quando já próximo carregar no botão correspondente ao *target*, que irá levar o robô para a posição pretendida.

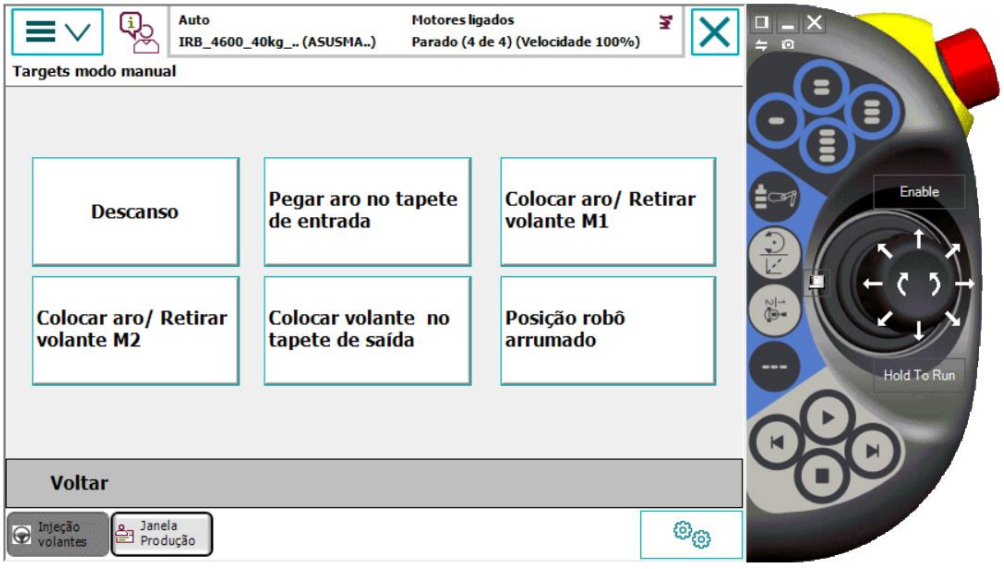


Figura 3-19 – Ecrã do modo manual que permite levar o robô para targets relevantes da célula robótica

Em relação aos menus Comandos Molde 1, Comandos Molde 2 e Comandos Cabeçal Injeção, possuem botões que permitem comutar manualmente a posição de cada um dos mecanismos, e também estão dotados de indicadores luminosos que, tal como no menu Comandos Garra, assinalam o estado dos mecanismos. Na Figura 3-20 pode ver-se o ecrã relativo aos comandos manuais do molde 1; para o molde 2 existe um ecrã em tudo idêntico. Na Figura 3-21 mostra-se o ecrã relativo aos comandos manuais do cabeçal de injeção.

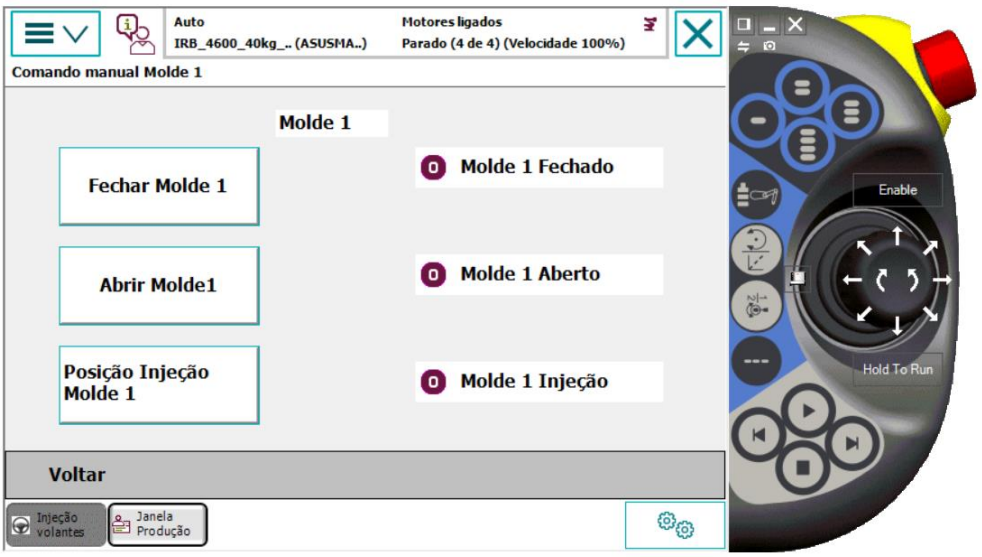


Figura 3-20 – Ecrã com comandos manuais do molde 1. Para o molde 2 existe um ecrã em tudo idêntico

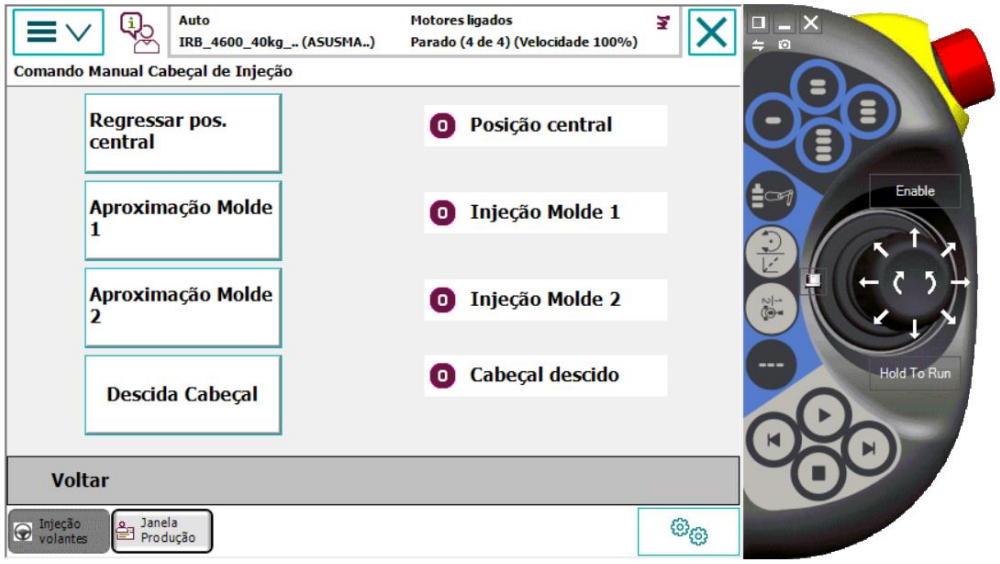


Figura 3-21 – Ecrã com comandos manuais do cabeçal de injeção

Quanto ao Modo Automático, o primeiro ecrã apresentado ao utilizador permite selecionar os moldes a serem utilizados, e também verificar a prontidão dos sistemas auxiliares

(Figura 3-22). Na implementação real será necessário criar em cada um dos autómatos dos sistemas auxiliares um sinal que indique a prontidão do mesmo, e ligá-lo ao controlador do robô. No presente trabalho de simulação, como todos os sistemas auxiliares são controlados pelo controlador virtual do robô, não faz sentido esta abordagem, pelo que a sinalização da prontidão é apenas feita para fins demonstrativos.

Caso se selecione o modo de funcionamento com dois moldes é a seguir dada a opção de escolher que molde será usado primeiro.

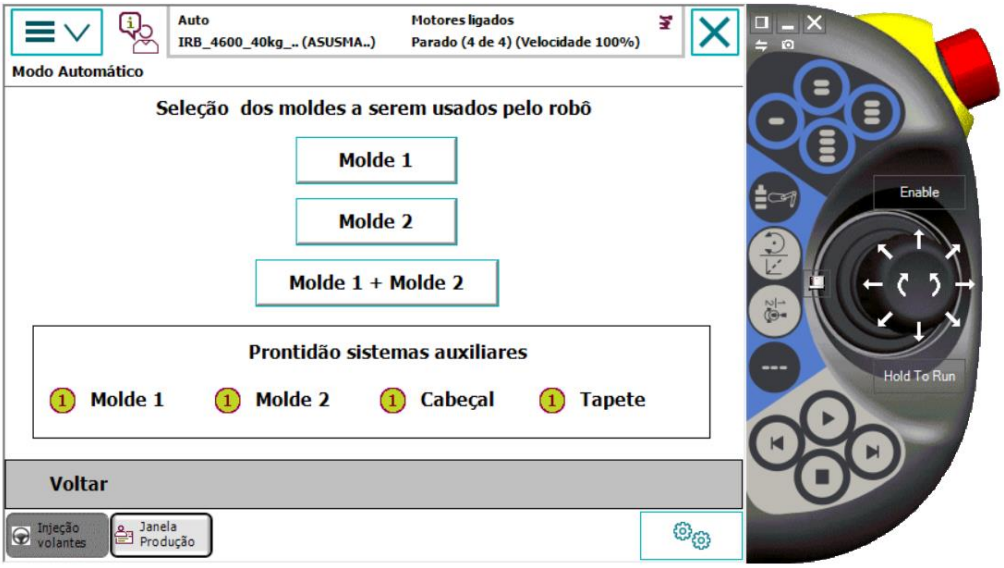


Figura 3-22 – Ecrã de seleção de moldes da interface desenvolvida para a célula robótica

Depois de escolhidos os moldes a operar, surge o ecrã de escolha de modelo de volante a produzir. É apresentada uma caixa de seleção, com a lista de modelos disponíveis. Cada modelo tem a si associado um tempo de cura, um percurso de limpeza, um percurso de aplicação de laca/desmoldante, e um tempo de injeção. Os parâmetros de cada volante são alteráveis através do código RAPID, editando as variáveis que guardam estes valores.

Ainda no mesmo ecrã está disponível a opção de produção em lote, que uma vez ativada permite definir o número de volantes a produzir. É também a partir deste ecrã que se dá a ordem de iniciar a produção. Este ecrã da interface está representado na Figura 3-23.

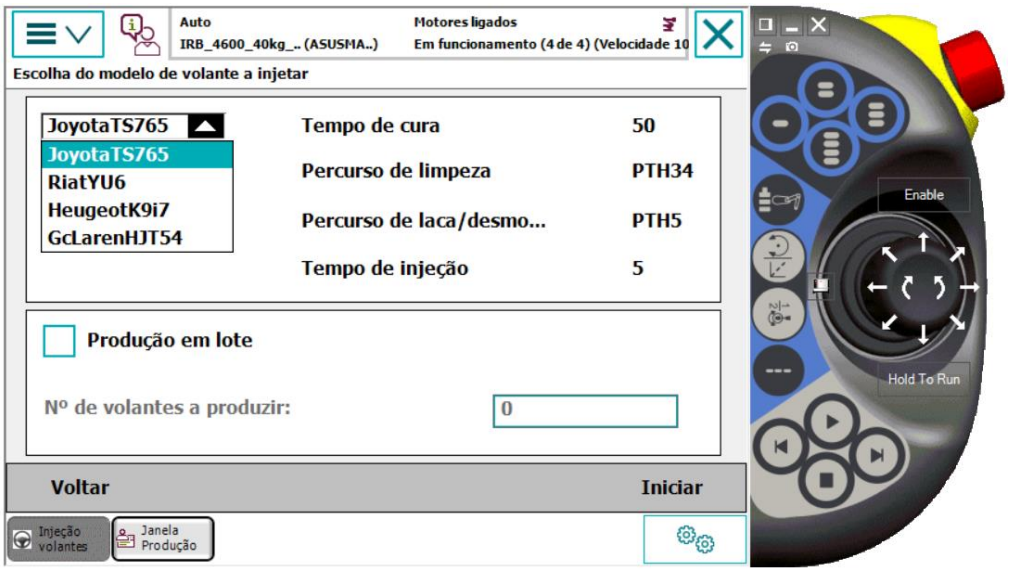


Figura 3-23 – Ecrã de seleção do modelo de volante a produzir

Quando a célula robótica se encontra em funcionamento é mostrado um ecrã (Figura 3-24) com o estado de funcionamento, onde são apresentados os principais dados relevantes: moldes em operação, modelo de volante em produção, volantes já produzidos, volantes a produzir. Está disponível um botão para realizar a paragem controlada da célula, como descrito no subcapítulo 4.4. A paragem não programada é realizada pelo botão físico STOP da consola. Este tipo de paragem permite retomar o funcionamento do programa onde foi interrompido.

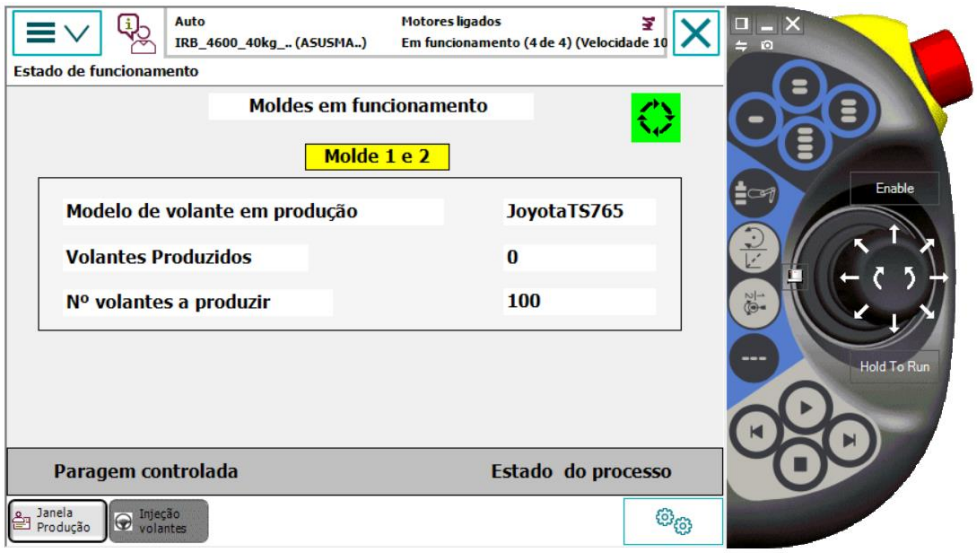


Figura 3-24 – Ecrã de monitorização do funcionamento da célula robótica

Está também disponível um botão “Estado do processo” que dá acesso a um ecrã onde é mostrada, para cada molde, a fase do processo atualmente em execução. Na Figura 3-25 está representado este ecrã de monitorização do processo.

Figura 3-25 – Ecrã de monitorização da fase do processo de injeção

Uma vez concluído o desenvolvimento da interface, a mesma foi usada para controlar a simulação no software RobotStudio. Foram então detetadas limitações ao correto funcionamento da simulação.

Nomeadamente, verificou-se que existem duas formas distintas de lançar a simulação no RobotStudio:

- Através do separador *Simulation*, usando o botão "*Play*". Lançando a simulação desta forma, não é possível usar a consola para definir o funcionamento da célula em termos de moldes em funcionamento nem modelo de volante a produzir.
- Através da consola virtual do robô. Ao iniciar a simulação pela consola já é possível configurar a célula e monitorizar o seu funcionamento. No entanto, foram encontradas algumas limitações nesta forma de execução da simulação.

As limitações encontradas prendem-se com o funcionamento dos *Smart Componentes* quando a simulação é lançada pela consola, e estão descritas de seguida.

1. Tapete de fornecimento dos aros de volante

Para simular o fornecimento de aros de volante é usado um *Smart Component* denominado SC_Aro_Volante. No entanto, iniciando a simulação pela consola do robô este *Smart Component* não tinha o funcionamento desejado, já que o bloco LinearMover, que move o aro no tapete, não era simulado. Por essa razão foi necessário desenvolver outro *Smart Component*,

que é usado apenas quando a simulação é iniciada pela consola. O objetivo deste *Smart Component* é replicar o funcionamento do *Smart Component* usado inicialmente, através de uma lógica de funcionamento diferente. Assim, a simulação funcionará independentemente do modo de inicialização da mesma (pela consola ou pelo botão "Play" do separador *Simulation*).

O funcionamento do novo *Smart Component* baseia-se no bloco "Source", que gera uma cópia de um objeto num ponto definido previamente. Neste *Smart Component* são usados quatro blocos "Source", posicionados com igual espaçamento no tapete de entrada de volantes, e ainda três blocos "Sink", que apagam os objetos gerados pelos blocos "Source".

Os blocos "Source" são ativados sequencialmente no código RAPID, com pausas entre ativações, e depois de ser criada a cópia do objeto é ativado o bloco "Sink" para a eliminar. A Figura 3-26 representa a lógica de funcionamento deste *Smart Component*.

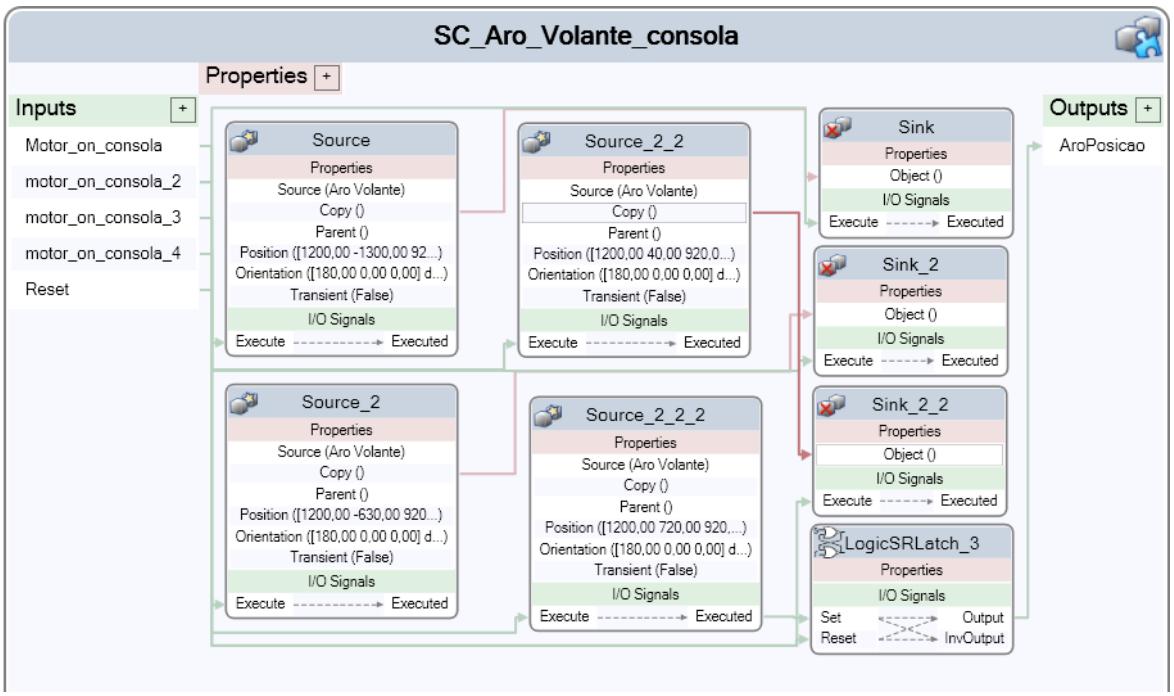


Figura 3-26 - Lógica de funcionamento do novo *Smart Component* do tapete de entrada de aros

2. Tapete de saída de volantes

O *Smart Component* responsável pelo tapete de saída de volantes também apresentou problemas de funcionamento quando a simulação é lançada pela consola, porque tal como o tapete de entrada de aros, usa um bloco de base chamado "LinearMover".

Por este motivo foi criado um *Smart Component* novo, destinado a ser usado apenas quando a simulação é lançada pela consola. Neste *Smart Component* foi usado o bloco de base "LinearMover2", que permite mover um objeto uma distância definida pelo utilizador. Constatou-se que este bloco-base funciona corretamente na simulação lançada pela consola. Depois de ser movido para o fim do tapete, o volante desaparece da simulação quando se ativa o bloco "Sink".

Na Figura 3-27 está representado a lógica de funcionamento deste novo *Smart Component*.

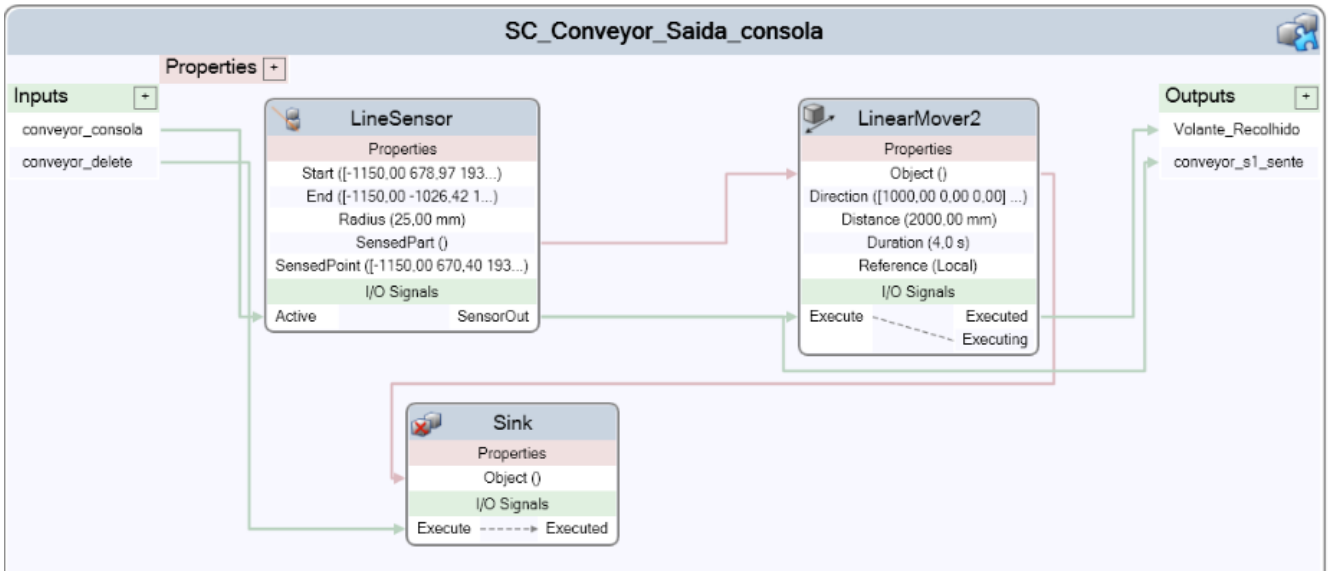


Figura 3-27 – Lógica de funcionamento do novo *Smart Component* do tapete de saída de volantes

3. Garra do robô

O *Smart Component* responsável pela simulação da garra do robô foi alterado devido à não atualização dos seus sinais de saída, o que fazia o programa não avançar quando era esperada uma alteração de algum desses sinais. Este problema ocorre apenas quando se inicia a simulação a partir da consola. Para contornar este problema foi necessário alterar a lógica do *Smart Component*, com a implementação de LatchRS antes de alguns dos sinais de saída.

Na Figura 3-28 está representada a lógica de funcionamento do *Smart Component* SC_Garra_Robot depois das alterações realizadas.

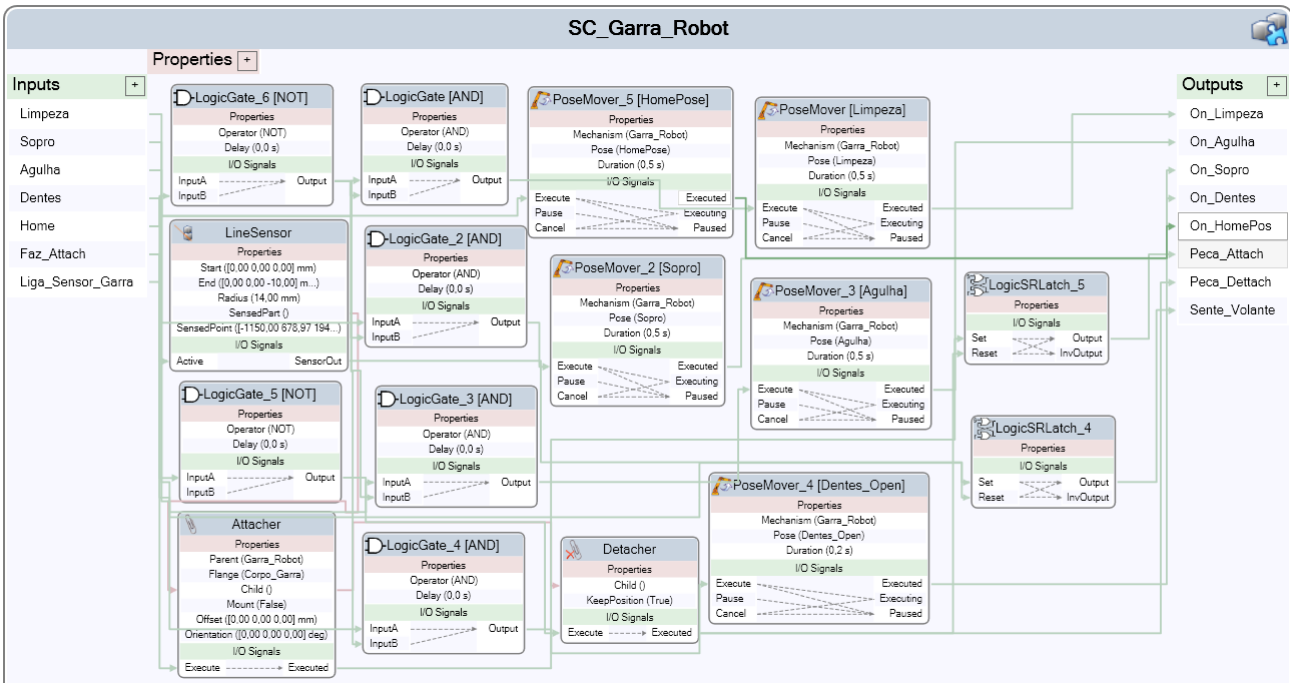


Figura 3-28 - Lógica de funcionamento do *Smart Component* da garra do robô

As LatchRS foram aplicadas ao sinal Sente_Volante, que indica a presença de um volante ou aro de volante nos dentes da garra, e ao sinal Peca_Attach, que indica a conclusão do procedimento de anexação do volante ou aro de volante à garra do robô, para ser transportado.

O *reset* da LatchRS que antecede o sinal Sente_Volante é feito pelo sinal Home, e o *reset* da LatchRS do sinal Peca_Attach é feito pelo sinal "Executed" do bloco Detacher.

4 Apresentação de resultados e análise comparativa das soluções

Neste capítulo é feita uma análise comparativa entre as soluções desenvolvidas, e entre essas e a solução existente aquando do início deste trabalho. Os critérios de comparação são o tempo de ciclo de cada solução, o atravancamento da célula, a configurabilidade do funcionamento, o custo da implementação, a facilidade de montagem e a área de implantação.

Para realizar uma comparação em iguais termos para todas as soluções, foram usados os mesmos percursos de limpeza, aplicação de desmoldante e de laca para a obtenção de todos os dados que se seguem. Foi ainda considerado um tempo de cura de 50 segundos e um tempo de injeção de 5 segundos.

É de notar que na solução de partida algumas rotinas tiveram de ser editadas, pois foi detetada a ocorrência de colisões, o que resultou num ligeiro aumento do tempo de execução dessas rotinas. Alguns exemplos desta situação são as rotinas de retirada de setas e colocação de aros.

4.1 Solução de partida

Como referido no subcapítulo 1.1, na solução de partida as várias etapas da injeção de um volante são executadas sequencialmente, não havendo sobreposição de operações. Esta situação deve-se à solução de programação adotada. Na Figura 4-1 está representado o diagrama temporal para a referida solução.

Nesta solução, a injeção dos 2 primeiros volantes demora cerca de 378 segundos. Este período de tempo é contado desde o início do funcionamento da célula até que é concluída a limpeza da parte superior do molde 2.

Durante o funcionamento cíclico, a injeção de 2 volantes demora aproximadamente 306 segundos. Este intervalo de tempo compreende todas as operações necessárias à execução dos 2 volantes, ou seja, é o tempo compreendido entre a aplicação do desmoldante na parte inferior do molde 2 até à limpeza da parte superior do molde 2 (ver Figura 4-1).

Com esta solução é possível obter uma cadência de produção de 23,5 volantes/hora.

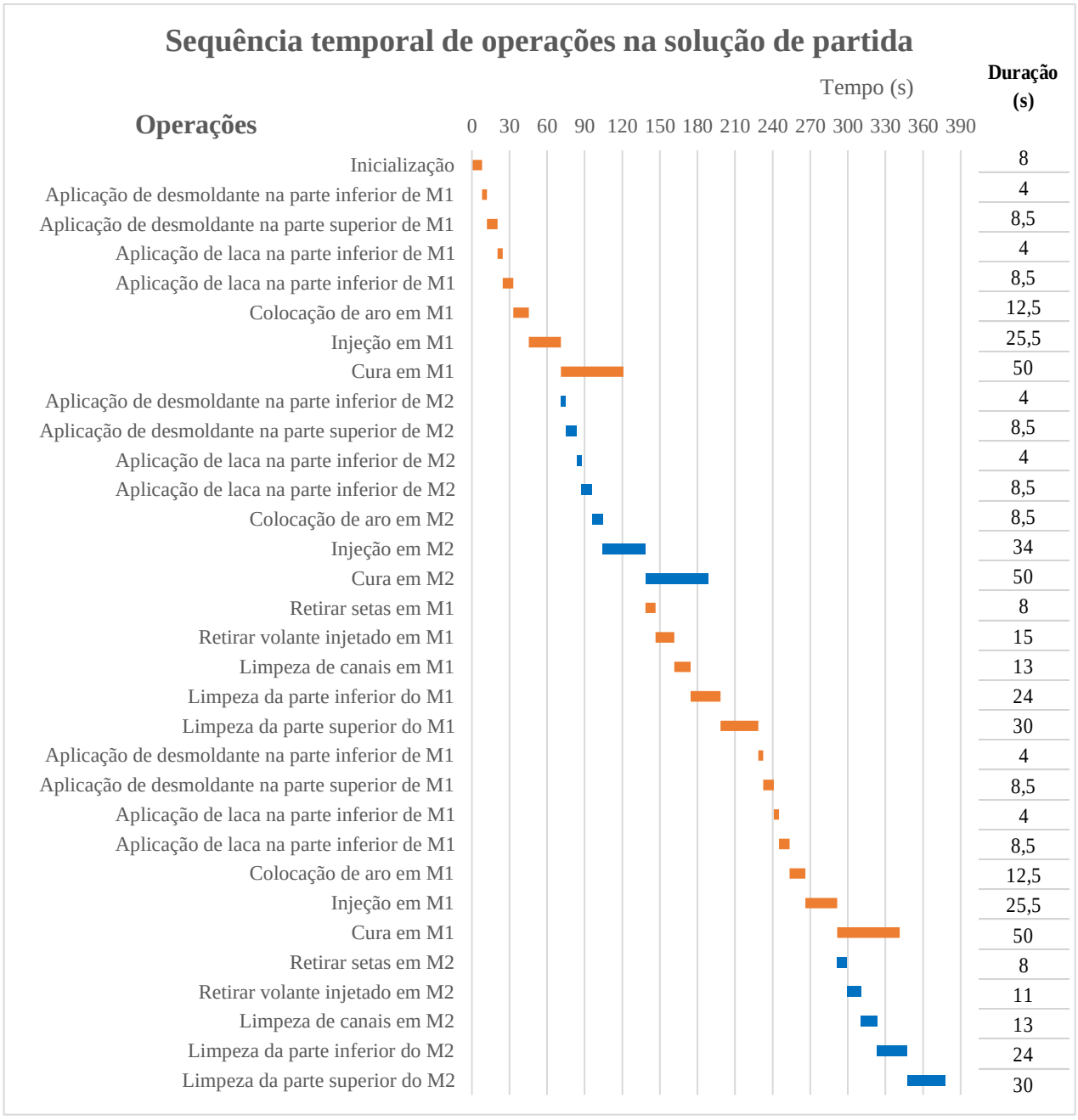


Figura 4-1 – Diagrama temporal do funcionamento da solução de partida

4.2 Solução 1

Na Figura 4-2 apresenta-se o diagrama temporal para a proposta da solução 1, com o robô IRB 4600 no solo. Na programação desta solução já foi usada a metodologia descrita no subcapítulo 3.4, de tal forma que exista sobreposição de tarefas com vista à redução do tempo de ciclo.

Nesta solução, a injeção dos dois primeiros volantes é feita em cerca de 326 segundos. Já em funcionamento cíclico, a injeção de 2 volantes é feita em 287 segundos.

Em relação à solução de partida, é obtido um ganho de 14% na injeção dos dois primeiros volantes, e um ganho de 7% no tempo de funcionamento cíclico.

Com esta solução é conseguida uma cadência de produção de 25 volantes/hora.

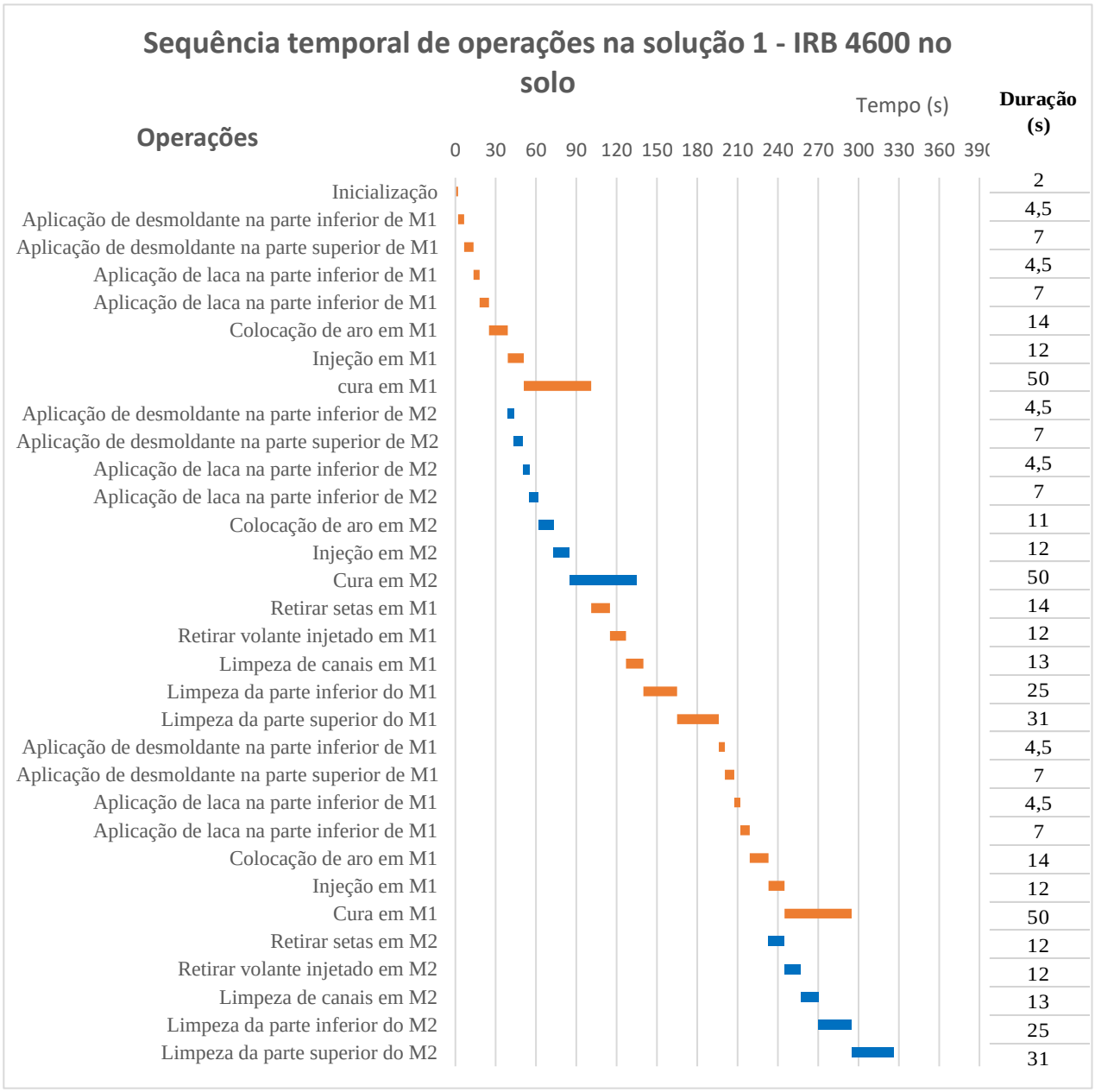


Figura 4-2 - Diagrama temporal do funcionamento da solução 1

4.3 Solução 2

Nesta solução, a injeção dos dois primeiros volantes é feita em cerca 322 segundos, como se pode ver na Figura 4-3. Em funcionamento cíclico a injeção de 2 volantes é feita em 287 segundos. Este intervalo de tempo compreende todas as operações necessárias compreendidas entre a aplicação do desmoldante na parte inferior do molde 2 até à limpeza da parte superior do molde 2.

Em relação à solução de partida, é obtido um ganho de 15% na injeção dos dois primeiros volantes, e um ganho de 7% no tempo de funcionamento cíclico.

Com esta solução é conseguida uma cadência de produção de 25 volantes/hora.

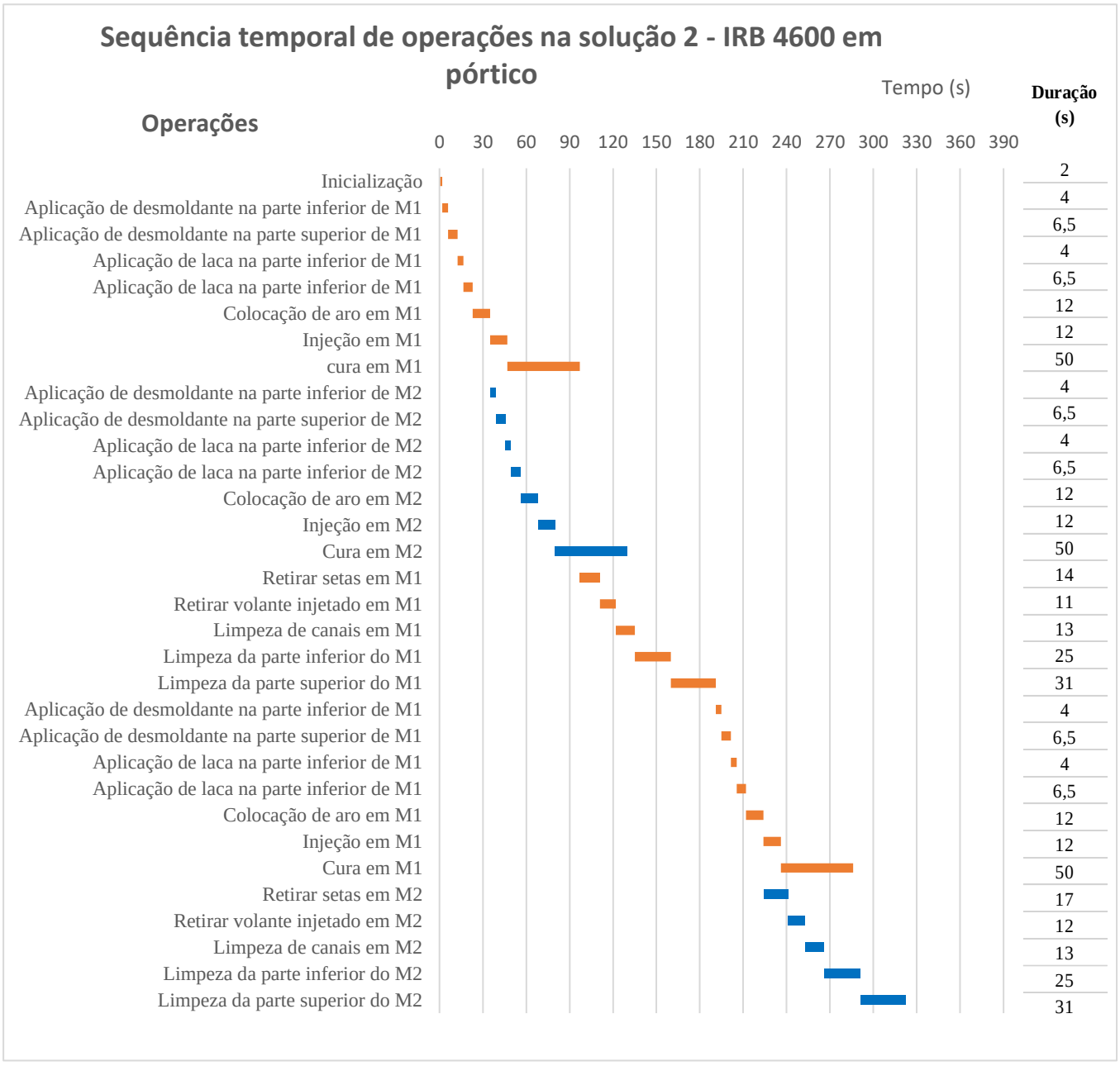


Figura 4-3 - Diagrama temporal do funcionamento da solução 2

4.4 Solução 3

Na proposta de solução 3, a injeção dos dois primeiros volantes é feita em cerca de 322 segundos, como se pode ver na Figura 4-4. Em funcionamento cíclico a injeção de 2 volantes é feita em 285 segundos.

Em relação à solução de partida, é obtido um ganho de 15% na injeção dos dois primeiros volantes, e um ganho de 7% no tempo de funcionamento cíclico.

Com esta solução é conseguida uma cadência de produção de 25 volantes/hora.

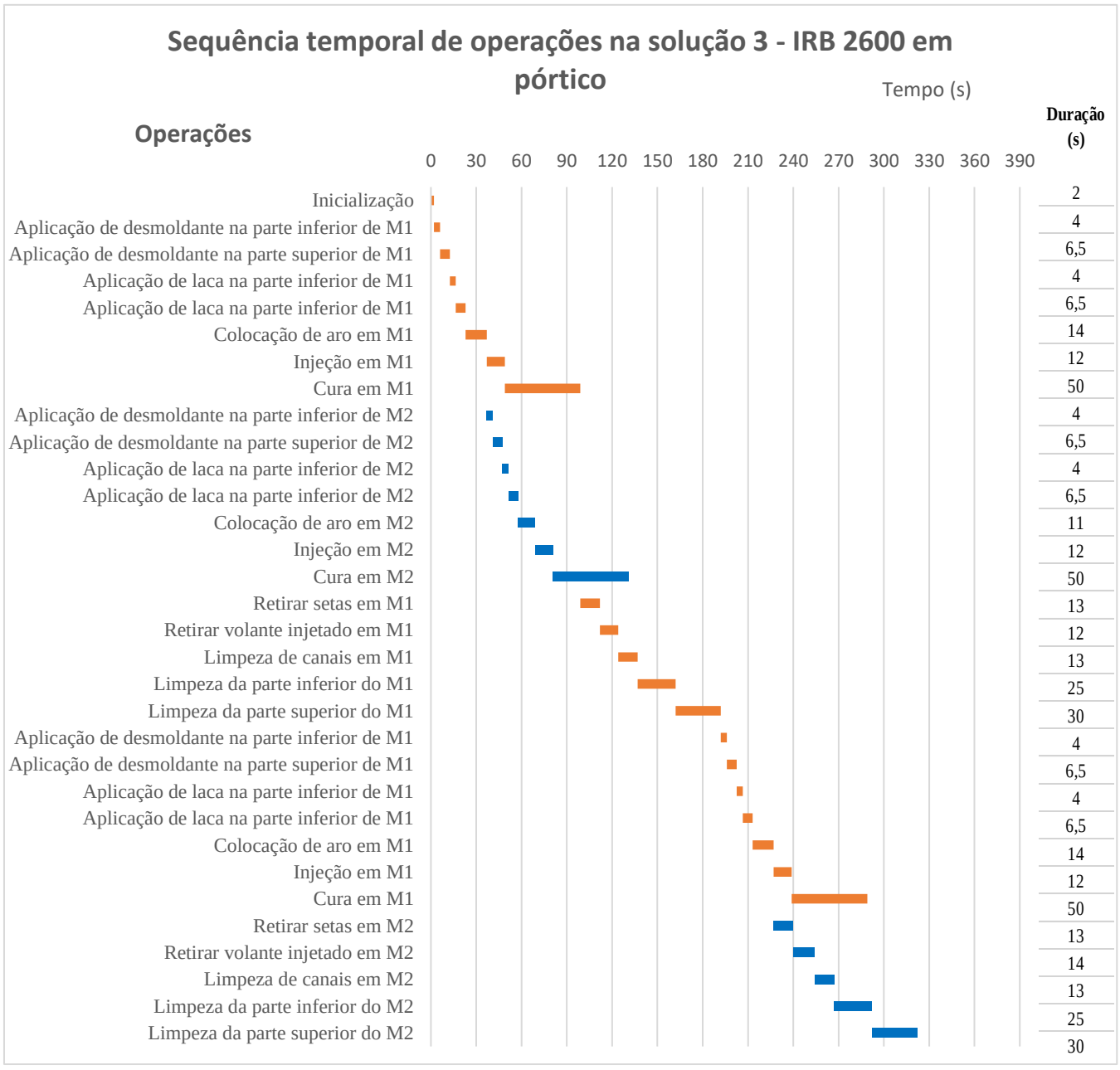


Figura 4-4 - Diagrama temporal do funcionamento da solução 3

4.5 Análise comparativa das soluções

De seguida é feita uma análise comparativa entre as várias propostas de *layout* para a célula robótica.

As três propostas de solução desenvolvidas neste trabalho apresentam tempos de produção muito semelhantes entre si, como foi visto nos subcapítulos 4.2 a 4.4. Comparando as novas soluções com a solução existente, foi conseguida uma diminuição do tempo de produção de 2 volantes, em funcionamento contínuo (cíclico), de cerca de 7%. Na injeção dos dois primeiros volantes foi obtido um ganho médio de tempo de 14%.

Em termos de acessibilidade da célula para operações de manutenção ou de mudança de molde, as soluções 2 e 3 são as que melhor acesso fornecem ao posto de injeção. Na solução 2, com o robô na posição "recolhido", está disponível uma altura mínima ao solo de 2200 mm. Na solução 3 essa altura é de 2250 mm. Para proceder a operações de manutenção ou mudança de molde é aconselhável comandar o robô para a posição programada no código destinada a esse fim, designada "recolhido", na qual não interfere com operários ou até máquinas que necessitem de entrar no espaço da célula. Nas outras duas soluções, a de partida e a solução 1, a acessibilidade é grandemente comprometida pela montagem do robô no solo, que atravanca grande parte do chão de fábrica usado pela célula, e dificulta a entrada e saída de máquinas, objetos ou operadores na célula, caso seja necessário.

Quanto à área de implantação necessária para efetivar cada uma das soluções, a solução de partida é a que necessita de menor área. A solução 1 necessita de ligeiramente mais área de chão de fabrica, devido ao posicionamento do tapete de entrada de aros. As soluções 2 e 3 são as que exigem uma maior área de implantação, devido à montagem do pórtico.

Em termos de custo de implementação, as soluções 2 e 3 serão as mais dispendiosas já que acarretam a montagem de um componente adicional, o pórtico. Pela mesma razão, estas duas soluções terão uma montagem mais laboriosa, demorada e dispendiosa.

Na Tabela 3 resumem-se as características das várias soluções comparadas neste capítulo.

Tabela 3 – Comparação das diferentes soluções propostas para o *layout* da célula robótica

	Solução Existente	Proposta Solução 1	Proposta Solução 2	Proposta Solução 3
Tempo de injeção dos 2 primeiros volantes (segundos)	378	326	322	322
Tempo de injeção de 2 volantes em funcionamento cíclico (segundos)	306	287	287	285
Cadência de produção em funcionamento cíclico (volantes/hora)	23,5	25	25	25
Acessibilidade da célula robótica para manutenção/ mudança de molde	Fraca	Fraca	Boa	Boa
Área de implantação	13,6 m^2	15,0m^2	23,9 m^2	23,9 m^2
Custo de equipamentos	+	+	+++	++
Custo da montagem	+	+	++	++

5 Conclusões e perspetivas de trabalho futuro

Depois de uma análise detalhada da solução existente para esta célula robótica, foram identificadas as áreas que seriam alvo de otimização neste trabalho. A otimização foi focada principalmente no posicionamento dos componentes da célula, na programação da mesma e no desenvolvimento de uma interface Homem-Máquina que permita configurar e monitorizar o funcionamento da célula. Com este trabalho pretendeu-se obter uma solução com menores tempos de produção, funcionamento configurável ao nível dos moldes a usar e dos modelos de volante a produzir, e ainda uma disposição da célula robótica que facilite operações de manutenção e mudança de moldes.

Começou-se por desenvolver três propostas de *layout* para a célula robótica, alterando o posicionamento dos componentes da mesma, como o robô e tapetes rolantes. A montagem do robô em pórtico foi estudada. Foi ainda desenvolvida uma solução com um modelo de robô de menores dimensões do que o inicial, com menor capacidade de carga, por forma a avaliar aplicabilidade de um robô de menor custo para a aplicação em causa.

Relativamente à programação da célula foram realizadas várias melhorias. Implementou-se uma lógica de programação multitarefa, de forma a possibilitar a execução de várias operações em simultâneo, nomeadamente movimentos do robô e atuação de *Smart Components*. Foi desenvolvido o código necessário para permitir a seleção dos moldes a operar, bem como selecionar o modelo de volante a produzir. É possível adicionar novos modelos de volante facilmente, bastando introduzir no código os parâmetros característicos do novo modelo, e adicionar os percursos de limpeza e aplicação de desmoldante usando os *templates* criados para o efeito. Um modo de produção por lotes foi implementado. Para que as exigências de produção fossem respeitadas, foram ainda adicionados temporizadores para o tempo de cura dos volantes. Foram ainda ajustados os percursos nas três soluções propostas para acomodar a mudança de localização dos componentes e para corrigir imperfeições detetadas na programação original.

Por fim foi desenvolvida a interface Homem-Máquina para o controlador do robô. Esta interface tem como objetivos configurar, operar e monitorizar a célula robótica. A interface

dispõe de um modo manual, no qual permite controlar vários sistemas da célula, tais como o robô, cabeçal de injeção, moldes e garra. No modo automático é possível definir os moldes a usar, o modelo de volante a produzir e definir a dimensão do lote a produzir. Para monitorizar o funcionamento estão disponíveis dois ecrãs, um com os dados de produção, tais como volantes já produzidos, modelo em produção, entre outros, e o outro ecrã com um diagrama que mostra a etapa de produção em cada molde.

Os resultados obtidos com este trabalho foram uma melhoria em termos de tempo de produção de dois volantes de 7%. As diferenças entre as várias soluções propostas, em termos de tempo de produção, são praticamente inexistentes.

Relativamente à acessibilidade da célula robótica para operações de manutenção, conseguiu-se uma melhoria significativa com as soluções 2 e 3, onde o robô foi montado em pórtico, deixando o centro da célula desimpedido.

Concluiu-se que é possível utilizar um robô de menor capacidade de carga e alcance, o que deve representar uma diminuição do custo associado ao robô, relativamente ao robô utilizado na solução inicial. Assim, de entre as três soluções propostas, recomenda-se a adoção do layout da solução três, no pressuposto de haver disponibilidade de área de implantação.

A nova programação desenvolvida permite um funcionamento configurável através da interface do controlador do robô. É possível selecionar os moldes a operar, o modelo de volante a injetar, a dimensão do lote a produzir.

A ferramenta de programação e simulação RobotStudio revelou-se de fundamental importância para os desenvolvimentos efetuados. A sua funcionalidade de verificação de alcance foi extensivamente utilizada para desenvolver as propostas de layout. Foi também de grande relevo o uso de *Smart Components* para simular, em ambiente virtual, o comportamento dos sistemas presentes na célula robótica, permitindo obter uma representação muito aproximada da célula real.

Tendo em vista o melhoramento das soluções aqui desenvolvidas, são propostos os seguintes tópicos relativamente a trabalhos futuros:

- Realizar um levantamento, na fábrica da TRW, do hardware usado nos sistemas auxiliares, para de seguida estudar a adequação da programação desenvolvida neste trabalho à implementação real da célula;
- Implementar na programação a possibilidade de produzir em simultâneo dois modelos de volante diferente, um em cada molde. Para desenvolver esta funcionalidade é necessário projetar um sistema de identificação dos diferentes aros

de volante, de tal forma que o robô selecione, de entre os vários aros em espera no tapete, o aro correto para o modelo de volante a produzir de seguida. De notar que esta funcionalidade pode ter interesse quando se esteja a produzir pequenas séries de volantes, e não compromete a cadência de produção, desde que os tempos de produção dos dois modelos de volantes sejam idênticos.

Referências

1. Robinson, A., "The History of Robotics in Manufacturing", consultado em Junho de 2017. Disponível em: http://cerasis.com/2014/10/06/robotics-in-manufacturing/
2. Abreu, P., "Robótica Industrial - Fundamentos da Robótica", FEUP, 2016.
3. Martins, R., "Conceção e simulação de um sistema robótico para operações de injeção de volantes", Dissertação de Mestrado, FEUP, 2016.
4. Barbosa, J., "Concepção e Simulação de Célula Robotizada para Operações de Acabamento", Dissertação de Mestrado, FEUP, 2010.
5. Viana, D., "Desenvolvimento de uma Solução Robótica para Operações de Acabamento de Solas de Sapatos", Dissertação de Mestrado, FEUP, 2010.
6. Neto, C., Desenvolvimento de componentes para a conceção de células robóticas em ambiente ABB RobotStudio", Dissertação de Mestrado, FEUP, 2016.
7. Güdel, "Catálogo Güdel", consultado em Março de 2017. Disponível em: https://gudel.picturepark.com/Website/Download.aspx?DownloadToken=e6797cb8-a056-4316-ad75-ec30adfce6d2&Purpose=AssetManager&mime-type=application/pdf#
8. ABB Robotics, "Ficha Técnica do robô industrial IRB 2600ID", consultado em Maio de 2017. Disponível em: https://library.e.abb.com/public/35e696f6317fd1be4825781600382e23/ROB0205EN_A.pdf
9. ABB Robotics, "Espaço de trabalho do robô ABB IRB 2600ID-15/1.85", consultado em Maio de 2017. Disponível em: http://www07.abb.com/images/default-source/robotics/irb-2600id-wr-2.jpg?sfvrsn=0
10. ABB Robotics, "Espaço de trabalho do robô ABB IRB 4600-40/2.55", consultado em Maio de 2017. Disponível em: http://www07.abb.com/images/default-source/robotics/irb-4600-wr-1.jpg?sfvrsn=0
11. ABB Robotics, "Technical reference manual RAPID overview", disponível no menu Help do software RobotStudio.

12. ABB Robotics, "Technical reference manual RAPID Instructions , Functions and Data types", disponível no menu Help do software RobotStudio.

13. ABB Robotics, "Application manual ScreenMaker", disponível no menu Help do software RobotStudio.